CREIME – A *C*onvolutional *R*ecurrent model for *E*arthquake *I*dentification and *M*agnitude *E*stimation

Megha Chakraborty [1,4], Darius Fenner [1,6], Wei Li [1], Johannes Faber [1,2], Kai Zhou [1,2,3], Georg Rümpker [1,4], Horst Stoecker [1,2,3,5], Nishtha Srivastava [1,4]

[1]Frankfurt Institute for Advanced Studies, 60438 Frankfurt am Main, Germany
[2]Institute for Theoretical Physics, Goethe Universität, 60438 Frankfurt am Main, Germany
[3]Xidian-FIAS international Joint Research Center, Giersch Science Center, 60438 Frankfurt am Main, Germany
[4]Institute of Geosciences, Goethe-University Frankfurt, 60438 Frankfurt am Main, Germany
[5]GSI Helmholtzzentrum für Schwerionenforschung GmbH, 64291 Darmstadt, Germany
[6]Johannes Gutenberg-Universität Mainz, 55122, Mainz,Germany

Key Points:

- We use a novel sequence-to-sequence mapping to train a deep learning model to detect an earthquake, pick the P-wave arrival and estimate its magnitude.
- The proposed model can perform reasonably well with 5 second windows containing only up to 2s of P-wave data.
- We show that our model can outperform traditional methods like STA/LTA and the existing deep learning models.

Corresponding author: Nishtha Srivastava, srivastava@fias.uni-frankfurt.de

Abstract

The detection and rapid characterisation of earthquake parameters such as magnitude are important in real time seismological applications such as Earthquake Monitoring and Earthquake Early Warning (EEW). Traditional methods, aside from requiring extensive human involvement can be sensitive to signal-to-noise ratio leading to false/missed alarms depending on the threshold. We here propose a multi-tasking deep learning model – the **C**onvolutional **R**ecurrent model for **E**arthquake **I**dentification and **M**agnitude **E**stimation (CREIME) that: (i) detects the earthquake signal from background seismic noise, (ii) determines the first P-wave arrival time and (iii) estimates the magnitude using the raw 3-component waveforms from a single station as model input. Considering, that speed is essential in EEW, we use up to two seconds of P-wave information which, to the best of our knowledge, is a significantly smaller data window compared to the previous studies. To examine the robustness of CREIME we test it on two independent datasets and find that it achieves an average accuracy of 98% for event-vs-noise discrimination and can estimate first P-arrival time and local magnitude with average root mean squared errors of 0.13 seconds and 0.65 units, respectively. We compare CREIME with traditional methods such as short-term-average/ long-term-average (STA/LTA) and show that CREIME has superior performance, for example, the accuracy for signal and noise discrimination is higher by 4.5% and 11.5% respectively for the two datasets. We also compare the architecture of CREIME with the architectures of other baseline models, trained on the same data, and show that CREIME outperforms the baseline models.

Plain Language Summary

The detection of earthquakes and rapid determination of parameters such as magnitude is crucial in Earthquake Monitoring and Earthquake Early Warning (EEW). Existing methods used to make such estimations are empirical and require expert analysts to define involved parameters, which is quite challenging. They are also sensitive to noise, which could lead to erroneous results. In this paper we propose a the **C**onvolutional **R**ecurrent model for **E**arthquake **I**dentification and **M**agnitude **E**stimation (CREIME) which is capable to detect an earthquake within 2 seconds of the first P-wave arrival and provides a first estimate for its magnitude. We test the model on two independent datasets to demonstrate its generalizability. CREIME successfully discriminates between seismic events and noise with an average accuracy of 98% and can estimate first P-arrival time and local magnitude with average root mean squared errors of 0.13 seconds and 0.65 units, respectively. We also show that CREIME can perform better than traditional methods like STA/LTA and previously published deep learning architectures in the context of rapid characterisation.

1 Introduction

According to its original definition (Richter, 1935) the magnitude of an earthquake is the logarithm of the maximum trace amplitude expressed in microns measured by a standard short-period torsion seismometer at an epicentral distance of 100km. It is one of *"the most important and also the most difficult parameters"* involved in real-time seismology (Jin et al., 2013) particularly since most magnitude scales such as local magnitude (m_L), body wave magnitude (m_B), surface wave magnitude (m_S) are empirical and saturate at different magnitude ranges (Chung & Bernreuter, 1981; Ekström & Dziewonski, 1988). This, coupled with the complexity of the nature of the geophysical processes affecting earthquakes, makes it very difficult to have a single reliable measure for the size of an earthquake (Kanamori & Stewart, 1978). Magnitude values measured in different scales may thus differ by more than 1 unit, particularly for extremely large events due to saturation effects (Howell Jr, 1981; Giardini, 1988; Geller, 1976; Kanamori, 1983). Even for the same magnitude scale, values reported by different agencies may differ by up to

0.5 units (Mousavi & Beroza, 2020). Traditionally, frequency-domain parameters such as predominant period τ_p^{max} (Nakamura, 1988; R. Allen & Kanamori, 2003), effective average period τ_c (Kanamori, 2005; Kuyuk & Allen, 2013; Jin et al., 2013) and amplitude domain parameters such as peak displacement (P_d) (Wu & Zhao, 2006; Kuyuk & Allen, 2013; Jin et al., 2013) calculated from the initial 1-3 seconds of P-waves have been shown to provide reliable estimates of (body wave) magnitudes through empirical relations. Such methods have been applied to Earthquake Early Warning (EEW) systems in Japan, California, Taiwan etc. (R. Allen et al. (2009) and the references therein). It has further been shown that the correlation of such parameters increases steadily upon increasing the duration of data used (Ziv, 2014). Thus, there is an *"inherent trade-off between speed and reliability"* (Meier et al., 2019).

Traditional machine-learning algorithms were "*limited by their inability to process data in its raw format*" (LeCun et al., 2015) and the need for hand-crafted features. This challenge has been overcome by the emergence of deep learning. Deep learning comprises hierarchical feature learning methods (LeCun et al., 2015), whereby several simple non-linear mathematical functions are applied to the raw data, to extract an increasingly abstract representation of the data at each level. It is the job of the deep learning model to learn the parameters of these functions. The advent of deep learning, coupled with the availability of large volumes of data and affordable computational power in the form of GPUs, have led to state-of-the-art results in image recognition (Krizhevsky et al., 2017; He et al., 2016), speech recognition (Mikolov et al., 2011; Hinton et al., 2012), and natural language processing (Peters et al., 2018; Collobert et al., 2011). In fields such as seismology, which have been data-intensive since their very origin and are witnessing an exponential increase in the volume of data (Kong et al., 2018), deep learning has proven successful in several tasks such as event detection(Perol et al., 2018; Z. Li et al., 2018; Meier et al., 2019; W. Li et al., 2022; Fenner et al., 2022) and phase picking (W. Zhu & Beroza, 2019; Mousavi et al., 2020; Liao et al., 2021; W. Li et al., 2021; Zhou et al., 2019), event location characterisation (Perol et al., 2018; Panakkat & Adeli, 2009; Kuyuk & Susumu, 2018), first motion polarity detection (Ross et al., 2018; Hara et al., 2019), among others.

A deep learning based approach for magnitude estimation was presented by Mousavi and Beroza (2020). The model presented in that paper focuses on estimating the magnitude for an earthquake waveform, using a window length of 30 seconds that includes both the P- and S-wave information. The input to the model are earthquake traces, and event-vs-noise discrimination and first P-arrival are not included in its goals. The use of deep learning facilitates the learning of the most relevant features directly from the waveform. This approach suffers from under-estimation at high magnitudes as these magnitudes are rare in nature and, hence, under-represented in the training data. In order to overcome this drawback we propose a two-pronged approach – resampling the data to get a more uniform magnitude distribution and penalising the underestimation of high magnitudes during model training. As already mentioned the model presented in Mousavi and Beroza (2020) uses S-wave information which makes it unsuitable for the purpose of rapid characterisation and EEW, where the information from the faster P-waves is leveraged to issue a warning before the slower, and more devastation S-waves hit the surface (Cremen & Galasso, 2020)).

In this paper we present a novel approach to achieve multi-tasking **C**onvolutional **R**ecurrent model for **E**arthquake **I**dentification and **M**agnitude **E**stimation (CREIME), which can simultaneously perform earthquake identification, local magnitude estimation and first P-wave arrival time regression solely based on 1-2 seconds P-wave recording. Unlike J. Zhu et al. (2021) which uses a set of twelve features extracted from 3 seconds of data to perform magnitude estimation, CREIME is end-to-end using a combination of Convolutional and Recurrent neural network to extract features directly from the raw waveform. The motivation for using such a small duration of P-wave data lies in its po-

tential utility in applications such as rapid earthquake characterisation for EEW systems (R. Allen et al. (2009); R. M. Allen and Melgar (2019) and references therein). While multiple-station based approaches are generally more robust and reliable, single station approaches are faster and therefore can be more useful in places where human settlements may lie very close to the earthquake epicenter, such as Southern California.

The model presented here can be seen as a prototype that can be adapted into EEW systems and has a potential to provide reliable first estimates. We demonstrate the robustness of our model, by testing it on two datasets. It is ensured that these datasets have no overlap in terms of the traces they contain to assert the generalizability of the model. We also compare the effects of using different types of ground motion data as the input to the model. As a final step we test the model on S-wave arrivals which are not encountered by the model during training, to verify that S-wave arrivals from low magnitude events do not get wrongly identified as P-arrivals for high magnitude events. This implies that the model can easily be adapted on real time data.

2 Data

2.1 STEAD

The data used to train and test CREIME has been obtained from the STanford EArthquake Dataset (STEAD) (Mousavi et al., 2019). It is a high-quality benchmarked global dataset of labelled seismograms which have been detrended, bandpass filtered between 1.0-40.0 Hz and resampled to 100 Hz. There are a total of 7 different types of instruments in which the data has been recorded, of these, 99.5% are either high-gain broad band or extremely short period. Each seismogram is of duration 1 minute and is represented in the form of NumPy arrays (Harris et al., 2020) of dimensions 6000×3. All earthquake waveforms are associated with local earthquakes with epicentral distance no greater than 350km. The metadata includes 35 attributes for each earthquake waveform and 8 attributes for each noise waveform.

For the sake of uniformity in magnitude, of the 23 different magnitude scales in which earthquakes are reported, we only choose events for which the magnitudes are reported in the 'ml' scale, i.e., local magnitude as these events constitute the majority (above 70%) of the dataset. To ensure that extremely noisy data is left out from the training and testing process only waveforms with a signal-to-noise ratio (provided in the metadata) above 10 dB are used (similar to Mousavi and Beroza (2020) where 20 dB is the cutoff signal-to-noise ratio). The noise and earthquake traces are roughly divided in the ratio 60:10:30 for training, validation and test sets. A total of 32,356 traces are used for training. For earthquake waveforms, it is made sure that all traces associated with one earthquake event are present in only one of the aforementioned three sets with the help of the 'source_id' attribute from the metadata. For noise waveforms, traces corresponding to a particular station can be present in only one of the three sets. This ensures that the test dataset is "truly unseen" to the model and hence, can give a reliable evaluation of the model's performance.

In accordance with the Gutenberg-Richter power-law (Gutenberg & Richter, 1944), high magnitude earthquakes are rare in nature. This power-law is reflected in the dataset as well (with a magnitude of completeness around 1-1.5). The distribution of magnitudes in the original dataset is similar to that of the testing data shown in Figure 1. This kind of imbalance in the distribution of the target variable in a regression problem tends to bias the model's performance towards lower magnitudes (<2.5) (Krawczyk, 2016) as observed in Mousavi and Beroza (2020). So, to make sure that the model can perform a reliable estimation over all magnitude ranges, we perform random under-sampling up to magnitudes of 4.0 and random over-sampling for magnitudes above 4.5. For this, different rates (chosen by trial and error) of undersampling or oversampling (achieved by

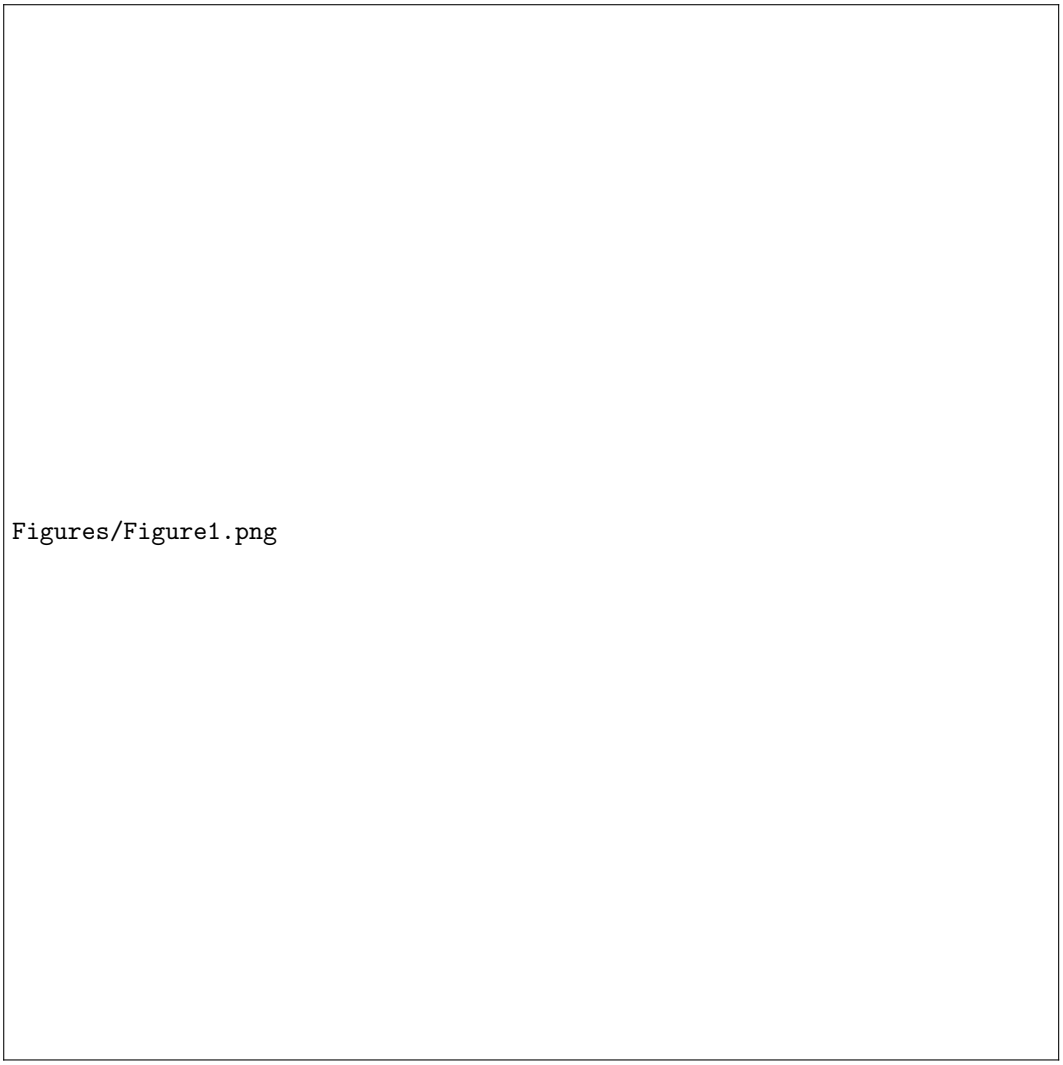


Figure 1. Distribution of magnitudes in training data (in slate blue) and chunk of STEAD(Mousavi et al., 2019) data used for testing (in orange). Note that the y-axis on the left corresponds to the training data distribution and that on the right corresponds to the test data distribution. While random undersampling and oversampling are applied to different magnitude ranges for training data in an attempt to get a uniform distribution, the original magnitude distribution of the test dataset is retained.

using windows with different starting time between 312-412 samples before P-arrival time) are applied to different magnitude ranges on the training and validation sets. This results in a training set with a magnitude distribution as shown in Figure 1. No such augmentation is applied to the test set (Figure 1) to retain the real world distribution of earthquake magnitudes encountered by the system. Furthermore, for training and validation, the number of noise traces chosen is exactly equal to the number of event traces.

2.2 INSTANCE

We further test our model on the INSTANCE dataset (Michelini, Cianetti, Gaviano, Giunchi, Jozinovic, & Lauciani, 2021), which is a recently published dataset comprising

1.2 million three-component waveform traces and 130,000 noise traces, each with a duration of 2 minutes, recorded primarily by the Italian National Seismic Network (network code IV). Corresponding to each trace 100 metadata, including magnitude and P-wave arrival sample, are provided. To make sure that there is no overlap with the training data, we exclude data from stations that are part of the STEAD dataset. We choose only traces for which magnitudes are provided in the 'ML' scale. For a fair evaluation of our model, we use only those traces with a single event and with distance and depth each within the corresponding maximum value present in the training data. Once again, traces with signal-to-noise ratio lower than 10 dB are not used. This leaves us with 135,347 traces corresponding to events between April 2005 to January 2020 and having a magnitude distribution as shown in Figure B1 in the appendix. The preprocessing steps for this data are very similar to those of the STEAD data except the bandpass filtering, so we apply a bandpass filter between 1.0 to 40.0 Hz using the *bandpass* function from *obspy.signal.filter* (Beyreuther et al., 2010).

3 Methodology

We use supervised learning (Chollet, 2017, Chapter 4) in this work to achieve earthquake identification and magnitude estimation, together with P-arrival time regression, based upon short records of P-wave data. The local magnitude (which is provided in the metadata for both the STEAD and INSTANCE datasets) or Richter scale magnitude (Richter, 1935) has the form:

$$M_L = \log A - \log A_0 + S \tag{1}$$

where, A is the peak horizontal amplitude measured on a Wood-Anderson seismograph, and A_0 and S are empirically determined distance and station correction terms derived from amplitude-distance relations representing attenuation and site functions respectively. While the peak amplitude can be directly obtained from the input data as we do not apply normalisation, it is expected that the model will learn the distance parameters, which are not provided explicitly, from the frequency content of the data itself. All three components are provided, to facilitate the learning of site effects (a similar approach has been followed by Mousavi and Beroza (2020)). We provide the data in units of 'counts' and do not perform instrument corrections, which gives the advantage that the analysis can be done in real-time.

A sequence-to-sequence approach is developed – the input to our model being 512 samples (5.12s) from 3 channels and the output is an array of the same length (512 samples). The data window for earthquake waveforms is chosen in such a way as to include 1 to 2 seconds of P-wave data, preceded by pre-signal noise (for noise waveforms the window has 512 samples of noise). This type of windowing allows the model to learn the noise characteristics (Münchmeyer et al., 2020). The Y-label for each X is a 512 $\times$ 1 array. These values are defined as follows:

$$y_i = \begin{cases} M & \text{if } i \geq i_p \\ -4 & \text{otherwise} \end{cases} \tag{2}$$

where M is the magnitude of the event and i_p denotes the P-arrival sample (in case of earthquake waveforms). The value -4 representing noise is arbitrary and chosen empirically by testing model performance on the validation data. The use of an arbitrary negative number to represent noise was explored by Yanwei et al. (2021). An example of this labelling for event and noise data is shown in Figure 2a. We have also tried modifying the final layer of the model to output two numbers corresponding P-arrival sample and magnitude instead of a sequence, similar to the approach of Yanwei et al. (2021) (not shown in the paper). However, our observation was that the sequence-to-sequence mapping approach leads to smaller errors.

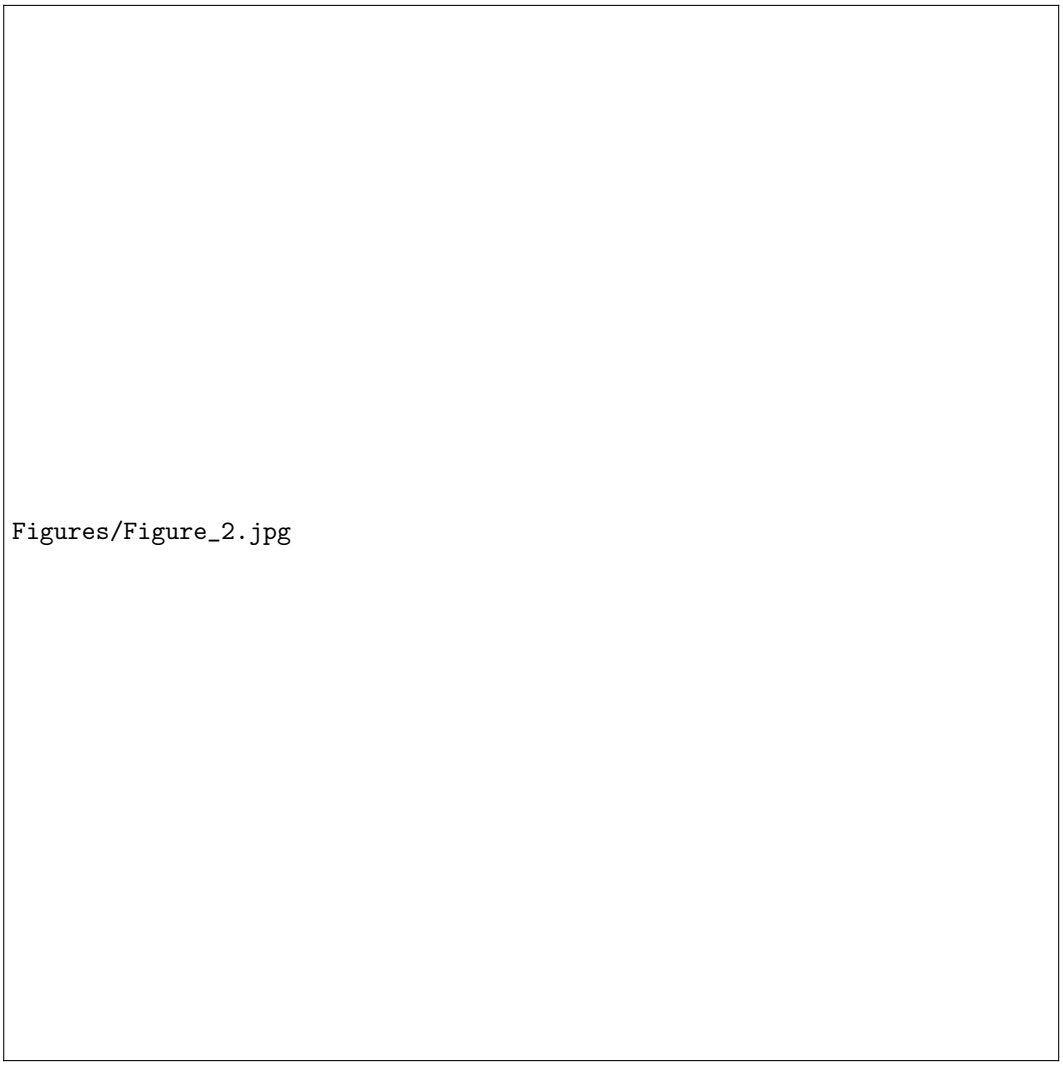


Figure 2. (a) Example of labelling for an event trace (left) and a noise trace (right); the label value is set to -4 for all samples before the P-arrival and the event magnitude for the P-arrival sample onward; for the noise trace it is set at -4 for all samples. (b) A schematic showing the architecture of the CREIME model; each convolution layer has a kernel size 16 and the number of filters are 32, 16 and 8; each Maxpooling layer reduces the dimension of the data by a factor of 4 and the Bi-LSTM layers have dimensions of 128 and 256 respectively.

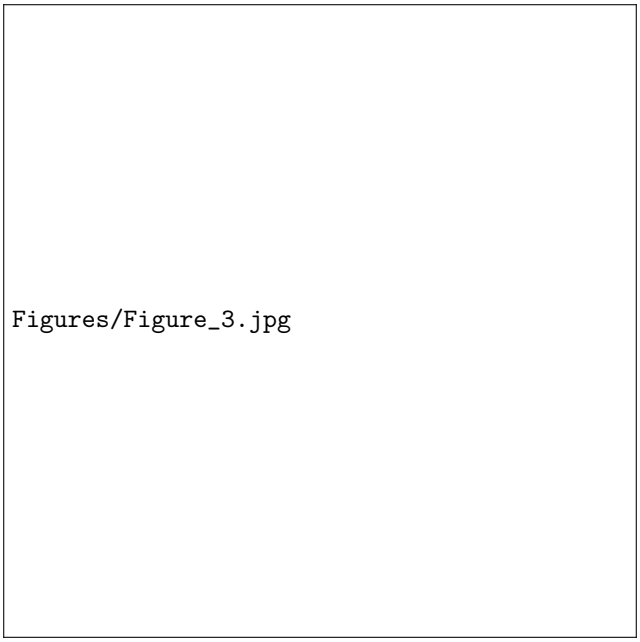


Figure 3. The variation of the training and validation loss as training progresses. The validation and training losses remain close to each other, which shows that the training is quite robust and there is no discernible overfitting.

The architecture of the CREIME model consists of three sets of 1D Convolution (Kiranyaz et al., 2015) and Maxpooling (Nagi et al., 2011) layer followed by two bidirectional Long-Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) layers of dimensions 128 and 256, which is followed by the output layer of dimensionality 512 (Figure 2b). The convolutional and maxpooling layers are used to extract and retain the relevant features while downsampling the data volume. Bidirectional LSTMs are used because of their ability to detect temporal dependencies for sequential data such as earthquake waveforms. Each convolution has a kernel size 16, a stride of 1, and padding type "same"; the number of filters is 32, 16 and 8, respectively. Each maxpooling layer reduces the size of the data by a factor of 4. Unlike the approach in Lomax et al. (2019) we find the model performance to be better when we use the original data without any normalisation. The model has a total of 1,454,992 trainable parameters and is trained using RMS Propagation optimiser (Tieleman & Hinton, 2012), with a batch size of 512. The model is implemented using Keras (Charles, 2013). On an NVIDIA A100GPU the training process takes less than 1 second per epoch. Each hyperparameter, including the number of layers in the model was chosen through meticulous experimentation by running several iterations of training and subsequent testing on the validation data.

We use early stopping (Prechelt, 2012) during the training to prevent overfitting. The validation loss is monitored and the training stops if it does not reduce for 15 consecutive epochs. We have an initial learning rate of 10^{-3} and reduce it by a factor of 10 until it reaches 10^{-6} if the validation loss does not go down for 10 consecutive epochs. The model with the lowest validation loss is saved. With these conditions the model trains for 71 epochs. The training history (i.e. learning curve) is shown in Figure 3.

For the cost function, we customized a combination of three losses, as different loss functions proved to be working better for different tasks and for different ranges of magnitude. The weights were determined by a trial and error method.

1. Mean Squared Error (MSE) with a weight of 40%: This is the average of squared values of errors corresponding to each data point in a minibatch. For k output values and a batch size n it has the form:

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{j=0}^{n-1}\frac{1}{k}\sum_{i=0}^{k-1}(y_{true}^{i,j} - y_{pred}^{i,j})^2 \tag{3}$$

   Here $y_{true}^{i,j}$ and $y_{pred}^{i,j}$ represents the true and predicted y values of the i-th sample for the j-th example in the minibatch, respectively.
2. Mean Absolute Error (MAE) with a weight of 40%: This is the average of absolute errors corresponding to each data point in a minibatch. For k output values and a batch size n it has the form:

$$\mathcal{L}_{MAE} = \frac{1}{n}\sum_{j=0}^{n-1}\frac{1}{k}\sum_{i=0}^{k-1}|y_{true}^{i,j} - y_{pred}^{i,j}| \tag{4}$$

3. Magnitude Estimation Loss with a weight of weight: 20%: As mentioned in the Introduction, we penalise the underestimation of magnitude, for high magnitude events (and overestimation for noise traces). To achieve this we define a third loss function. For k output values and a batch size n it has the form:

$$\mathcal{L}_{ME} = \frac{1}{n}\sum_{j=0}^{n-1}\alpha^j\frac{1}{k}\sum_{i=0}^{k-1}(y_{true}^{i,j} - y_{pred}^{i,j}) \tag{5}$$

   where,

$$\alpha^j = \begin{cases} \text{Event Magnitude}, & \text{for events} \\ -4, & \text{for noise} \end{cases}$$

As already mentioned, we utilise the output from our model to perform three tasks: discrimination between seismic event and noise, magnitude estimation, P-arrival sample detection. Based on a manual investigation of the output data and a subsequent testing on the validation dataset we used the following analysis to extract the desired parameters from the 512 sample sequence output by the model:

1. Predicted magnitude,

$$M_{pred} = \frac{1}{10}\sum_{i=(k-9)}^{k} y_{pred}^{i} \tag{6}$$

   where k is the number of samples, in our case, 512
2. Considering the first sample point in the data window as zeroth sample, we define

$$\text{P-arrival sample} = i_{pred}^{p} \text{ such that } y_{pred}^{i} > -0.5 \text{ for all } i \geq i_{pred}^{p} \tag{7}$$

3. If M_{pred} calculated by equation (6) is less than -0.5 then it is classified as noise.

The value -0.5 is chosen empirically based on the magnitude range of the data. For a detailed description of the metrics please refer to the Appendix A.

4 Results

The model was tested on a chunk of the STEAD dataset (Figure 1). Figure 4a shows the confusion matrix for noise/event classification; Figure 4b shows that the predicted magnitudes for noise data wrongly classified as event tend to be low (mostly ≤ 1) indicating that the possibility of false alarms caused by noise is low and events which are wrongly classified as noise are usually of low magnitude (≤ 2) indicating a low risk of

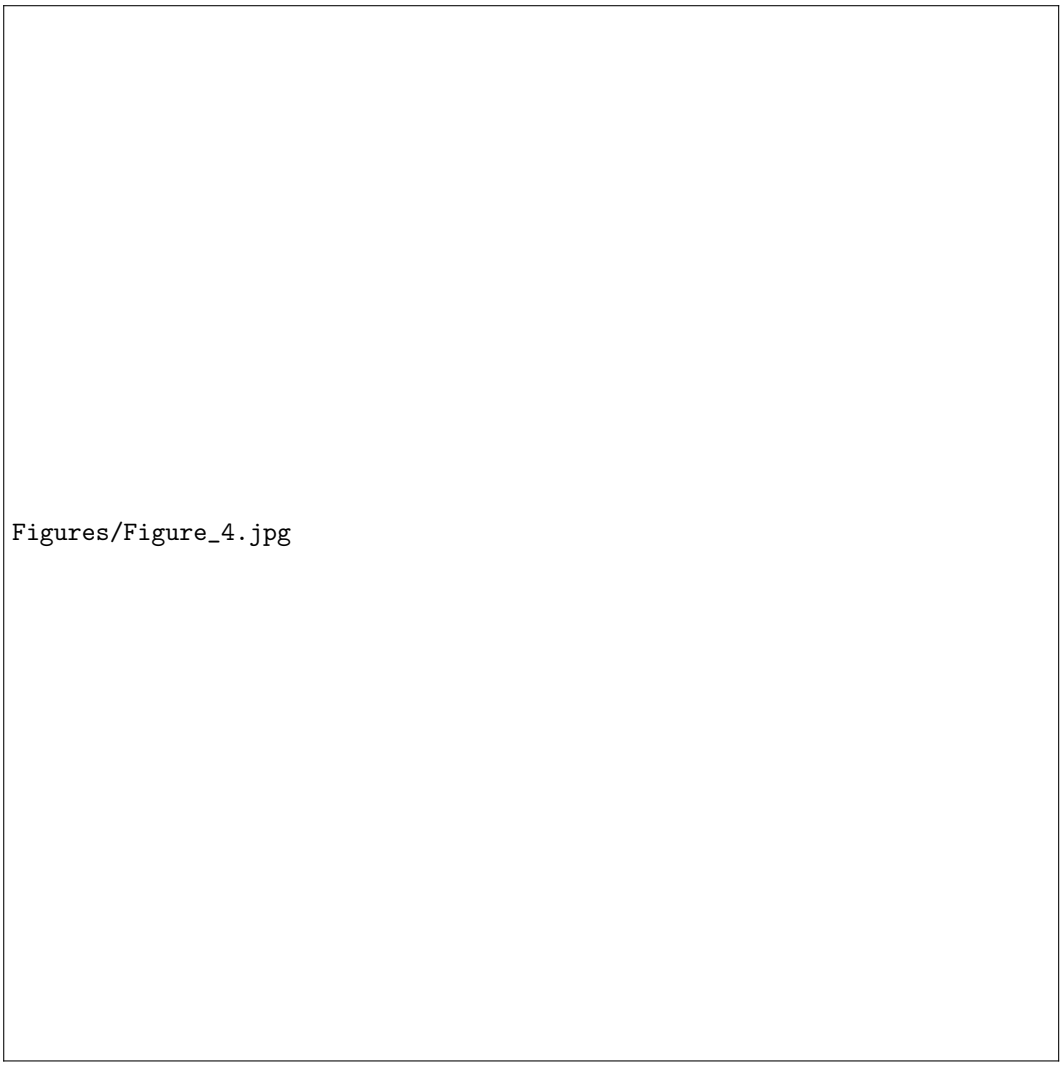


Figure 4. Analysis of CREIME model performance as a classifier on STEAD Data. It achieves an accuracy of 98.58%. The true magnitude of events misclassified as noise and predicted magnitude of events misclassified as noise tends to be low, which reduces the chance of missed or false alarms. (a) Confusion matrix for classifier performance. (b) Distribution of predicted magnitudes of noise misclassified as event and true magnitude of events misclassified as noise.

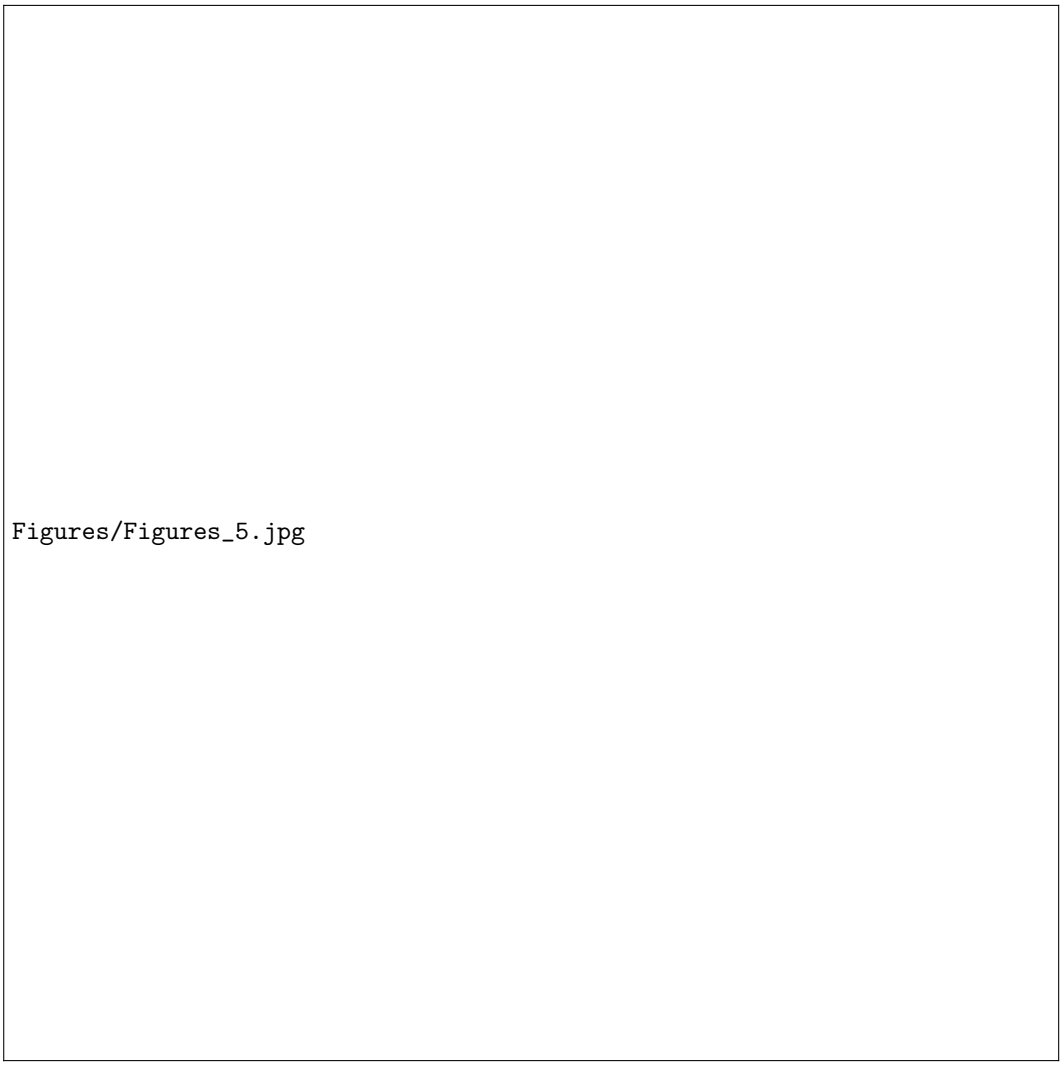


Figure 5. (a) Example of correct classification of an event trace (left) and a noise trace (right); one can see that the predicted magnitude for the event trace is very close to the true magnitude. (b) Example of incorrect classification of an event trace as noise trace(left) and a noise trace as an event trace (right); the event is a low magnitude one, and quite difficult to identify in this frequency range; the noise level in case of the noise trace is quite high gets classified as a low magnitude earthquake.

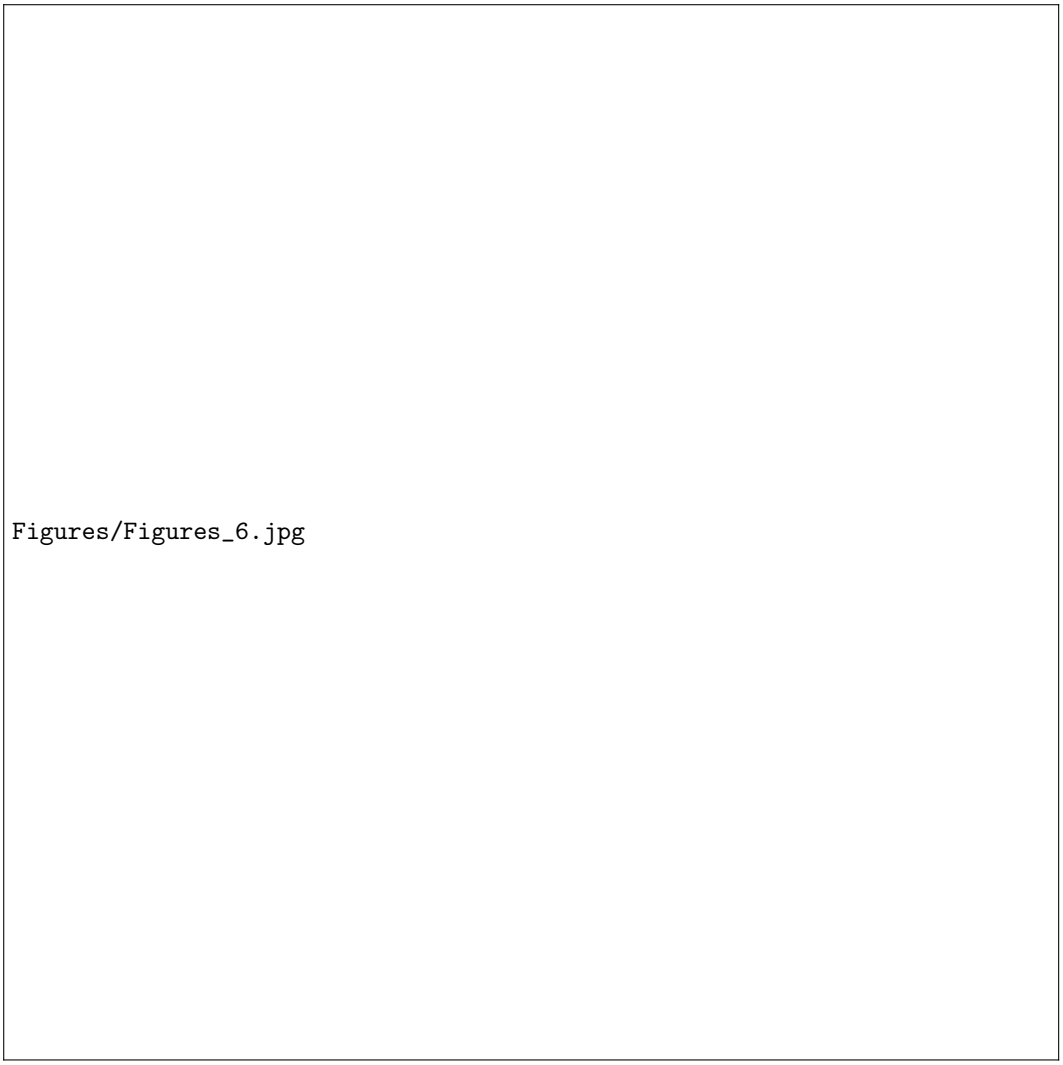


Figure 6. Analysis of model performance as a regressor on STEAD Data. The density plot shows that the highest density of points lies close to the zero error line; in spite of our penalization of under-estimation of high magnitudes, some under-estimation is observed above a magnitude of 5.5. In over 90% of the cases, the error in predicted magnitudes is less than 1 unit. (a) Relationship between true and predicted magnitude values. (b) Distribution of errors in predicted magnitudes. (c) Distribution of errors in P-arrival estimation.

false alarms, which is reassuring. Figure 5 shows examples of input and corresponding outputs for correctly and incorrectly classified traces.

The scatter plot for predicted versus true magnitudes is shown in Figure 6a. It is worth to note that for majority of the events (shown with higher relative density in the plot) the prediction reproduce well for the true magnitudes up to 5.5. For higher magnitudes events, some degree of underestimation is observed in spite of the penalty incorporated in the loss function. The result here, however, is an improvement over Mousavi and Beroza (2020), where magnitude underestimation starts to occur from a magnitude of 4. It should be taken into account that the data used to train the MagNet Mousavi and Beroza (2020) has a signal-to-noise ratio above 20 dB whereas, we use a lower threshold of 10 dB in our analysis. The histogram for errors in magnitude (Figure 6b) has a mean of -0.06 units, and a slight left skew, reflecting our penalisation of underestimation of magnitudes. The histogram for errors in predicted P-arrival (Figure 6c), is also unimodal, with a higher negative skewness indicating, that the P-arrival is more often predicted to be at a *later* time than it really is. The kurtosis for errors in P-arrival prediction is also much higher than that for magnitude prediction, indicating that errors in P-arrival predictions have a much narrower peak compared to errors in magnitude prediction. Similar results are observed for the INSTANCE dataset. We refer interested readers to Appendix B for the corresponding figures.

Comparison with other models

We compare our model with ones published in the papers listed below. It is important to note here, that the input data for the models in these studies differs from our data in terms of length, pre-processing etc. Therefore, for an unbiased comparison, all models have been retrained on the same training data that we use for our model. This is essentially a comparison between different architectures and not between the methodology presented by the respective authors.

1. MagNet (Mousavi & Beroza, 2020): This paper presents a deep learning model to perform only magnitude estimation using 30 seconds of data including both P and S phases. While both MagNet and CREIME use a combination of CNNs and bidirectional LSTMs, they differ significantly in the number of layers (MagNet uses 2 Convolutional layers and 1 bi-LSTM whereas CREIME uses 3 Convolutional layers and 2 bi-LSTMs), the model output(MagNet outputs the estimated magnitude and the aleatoric uncertainty whereas CREIME outputs a 512 dimensional array) and the choice of hyperparameters (such as number of filters in the Convolutional layers and dimension of LSTM). Unlike MagNet, CREIME does not use dropout layers. The only modification we make to the original architecture of MagNet, while re-training it, is to change the input shape from (3000,3) to (512,3). We then compare this model with CREIME in terms of estimation of event magnitudes.
2. CNN model for signal noise discrimination (Meier et al., 2019): The model presented in this paper originally takes 4s of data, starting 0.5 to 1.5 seconds before the P-arrival to discriminate between earthquake signals and noise. We train it on our data while keeping the architecture intact except a change in the input dimensions. Unlike the original paper, however, we do not impose a lower limit on the magnitudes of the events.
3. ConvNetQuake_INGV (Lomax et al., 2019): This model is inspired by the ConvNetQuake (Perol et al., 2018), and uses 10 seconds of data to perform multiclass classification to identify seismic events and characterise earthquake parameters such as magnitude, distance, depth and azimuth. While the original architecture uses 9 CNN layers, each downsampling the data by a factor of 2, we use only 8 (similar to Perol et al. (2018)) since the length of data in our case is almost half of that in the original paper. Further, in the last layer we use 31 classes for magnitude instead of 20 in the original paper giving a total of 32 nodes (one for sig-

Table 1. Comparison between the performance of CREIME model and other baseline models as a classifier for events and noise. CREIME model outperforms the other models.

		Metric						
			Precision (%)		**Recall (%)**		**F1-score (%)**	
Dataset	***Model Architecture***	**Accuracy (%)**	Event	Noise	Event	Noise	Event	Noise
STEAD	*CREIME Model*	**98.58**	**99.64**	**96.25**	**98.31**	**99.18**	**98.97**	**97.70**
	CNN Model	89.72	99.18	75.37	85.93	98.37	92.08	85.35
	ConvNetQuake_INGV	96.56	99.12	91.30	95.91	98.05	97.49	94.55
	STA/LTA Algorithm	94.08	96.03	89.70	95.43	91.00	95.73	90.34
INSTANCE	*CREIME Model*	**97.59**	**98.66**	**95.75**	**97.53**	**97.68**	**98.10**	**96.71**
	CNN Model	91.71	96.77	84.33	90.00	94.71	93.23	89.22
	ConvNetQuake_INGV	86.48	96.00	74.16	82.79	93.47	88.90	82.70
	STA/LTA Algorithm	86.03	90.87	78.49	86.81	84.66	88.79	81.46

Table 2. Comparison between magnitude estimation by CREIME model and other baseline models. The smallest errors are shown by CREIME model.

		Metric			
Dataset	***Model Architecture***	**Mean Error**	**St. dev. of Error**	**RMSE**	**MAE**
STEAD	*CREIME Model*	**-0.06**	**0.60**	**0.61**	**0.46**
	MagNet	-0.29	0.65	0.72	0.53
	ConvNetQuake_INGV	0.41	1.05	1.13	0.94
INSTANCE	*CREIME Model*	**-0.02**	**0.69**	**0.69**	**0.54**
	MagNet	-0.33	0.80	0.86	0.68
	ConvNetQuake_INGV	0.78	0.98	1.25	1.04

nal vs noise discrimination). To compare the magnitude regression performance with CREIME we take the predicted magnitude to be the arithmetic mean of the boundaries for the predicted class.

In addition to these deep learning models, we also compare our model with the Short-Term Average/Long-Term Average method (STA/LTA)(R. V. Allen, 1978), to evaluate the performance of our model in terms of classification and P-arrival time prediction. This is done by using the *classic_sta_lta* from *Obspy* (Beyreuther et al., 2010). The best set of parameters, determined on the basis of a grid search on the training data are: short-term window length = 20 samples (0.2s), long-term window length = 200 samples (2s), and trigger threshold = 4.0.

The performance metrics for the CREIME classifier in comparison with other classification models and the STA/LTA algorithm are summed up in Table 1. CREIME outperforms all the other architectures trained on the same data, and the conventional STA/LTA algorithm. The performance of the model in estimating magnitude and P-arrival time is summarised in tables 2 and 3 respectively. CREIME model outperforms MagNet and ConvNetQuake_INGV in terms of magnitude estimation. It also gives lower values for both RMSE and MAE compared to STA/LTA algorithm.

5 Discussion

We investigated the different factors that influence the results of our model. Figure 7a shows the variation of errors with the signal-to-noise ratio in the data. It is observed that the errors in magnitude and P-arrival time show highest density within ± 1 units and ± 0.1 seconds, respectively, and tends to be lower for higher signal-to-noise ratios.

Table 3. Comparison between CREIME model and STA/LTA method in terms of P-arrival picking. CREIME model outperforms STA/LTA

		Metric			
Dataset	***Model Architecture***	**Mean Error (s)**	**St. dev. of Error (s)**	**RMSE (s)**	**MAE (s)**
STEAD	*CREIME Model*	-0.05	**0.10**	**0.12**	**0.08**
	STA/LTA	**0.01**	0.37	0.36	0.18
INSTANCE	*CREIME Model*	-0.04	**0.13**	**0.14**	**0.09**
	STA/LTA	**0.01**	0.52	0.52	0.29

Figures/Figure_7.jpg

Figure 7. Factors affecting the error in estimation of magnitude and P-arrival times; errors in both magnitude and P-arrival are lower for higher signal-to-noise ratios; the magnitude of events seems to be under-estimated for higher hypocentral distances owing to their under-representation in the data. (a) Variation of errors with signal to noise ratio. (b)Variation of errors with hypocentral distance.

Figure 7b shows the variation of errors with hypocentral distance. We see that the errors tend to be close to zero over a wide range of hypocentral distances (up to 200km). There is a tendency for the model to underestimate the magnitude for higher hypocentral distances, which are under-represented in the training data. Both these figures are generated using STEAD data, and the corresponding figures for INSTANCE data can be found in the Appendix B.

We further looked into the effect of using different types of ground motion data as input (by removing instrument response), a summary of which can be found in Appendix C.

To make sure, that S-arrivals for low magnitude earthquakes do not get detected as high-magnitude events, we test the model on S-arrival data. We do not notice any systematic overestimation, only in 9% of the cases in the overestimation more than 1 unit. This means that our model can be applied to the incoming seismogram in real time for rapid characterisation. Comparing the performance of the CREIME model with our observations in Chakraborty et al. (2021), we find that providing data labels in the form of a series and including the first P-arrival information is beneficial for the model, in estimating the earthquake magnitude.

6 Conclusion

We present a novel deep learning model, CREIME, which successfully unifies the tasks of event and noise discrimination, P-arrival time estimation and magnitude estimation using a smaller window (up to 2 seconds) of P-wave data as compared to previously published models. The model in its current form, however, is restricted by the fact that was trained specifically on data windows where the P-wave arrival is between 3.12 and 4.12 seconds of the starting sample. This restriction can be overcome in a future version of the model by modifying the training dataset to include a wider range of arrival times. Nevertheless, this model can be seen as an important first step to a fully automated earthquake characterisation approach in real time. We show that it performs better than baseline models re-trained on the same duration of data. It also outperforms traditional event discrimination algorithms such as STA/LTA. We demonstrate the robustness of our model by testing it on two independent datasets, and show that it can provide reliable estimates over a wide range of hypocentral distances and signal-to-noise ratios. The model is designed to handle seismological waveform data in its raw format, which makes it very efficient in handling big data. Such models can also find their utility in smartphone applications to issue timely warnings to the public, as smartphone sensors have been shown to be capable of detecting seismic events (Kong et al., 2016).

Appendix A Metrics used for model evaluation

A1 Classification Metrics

We use different kinds of metrics to evaluate the classification and regression tasks. The performance of a classifier is often visualised with the help of a confusion matrix (Ting, 2017). The metrics we use to evaluate our model performance are described below. The abbreviations used are: TP: True positives

TN: True negatives

FP: False positives

FN: False negatives

- **Accuracy**: The accuracy of a classifier is the ratio of the number of correct predictions to the total number of predictions made by the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{A1}$$

- **Precision**: The precision of a classifier is the ratio of the number of correct predictions for a particular class to the total number of times that class is predicted.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{A2}$$

- **Recall**: The recall of a classifier is the proportion of the number of instances of a class in the data set that are correctly predicted.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{A3}$$

- **F1 Score**: By definition, there is an inherent trade-off between the precision and the recall of a classifier. Therefore, it is often worthwhile to look at the harmonic mean of the two. This metric is called the F1-score of the classifier.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A4}$$

A2 Regression Metrics

For the regression task, the following metrics will be used to measure the CREIME performance:

- **Mean Error**: This is the mean value of errors corresponding to each example in the data set.

$$\text{Mean Error, } \bar{\mathcal{E}} = \frac{1}{N}\sum_{i=0}^{N-1}\mathcal{E}_i = \frac{1}{N}\sum_{i=0}^{N-1} y^i_{true} - y^i_{pred} \tag{A5}$$

  where N is the total number of examples in the dataset.
- **Standard Deviation of Error**: This is the standard deviation of the errors in the predictions.

$$\text{Standard Deviation of Error, } \sigma_{\mathcal{E}} = \sqrt{\frac{\sum_{i=0}^{N-1}(\mathcal{E}_i - \bar{\mathcal{E}})^2}{N}} \tag{A6}$$

- **Root Mean Squared Error (RMSE)**: As the name says, this is the square root of the mean of squares of errors in prediction.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{N-1}\mathcal{E}_i^2}{N}} \tag{A7}$$

- **Mean Absolute Error**: This is the mean of the absolute values of the errors in prediction.

$$\text{MAE} = \frac{\sum_{i=0}^{N-1}|\mathcal{E}_i|}{N} \tag{A8}$$

Appendix B Model performance on INSTANCE Dataset

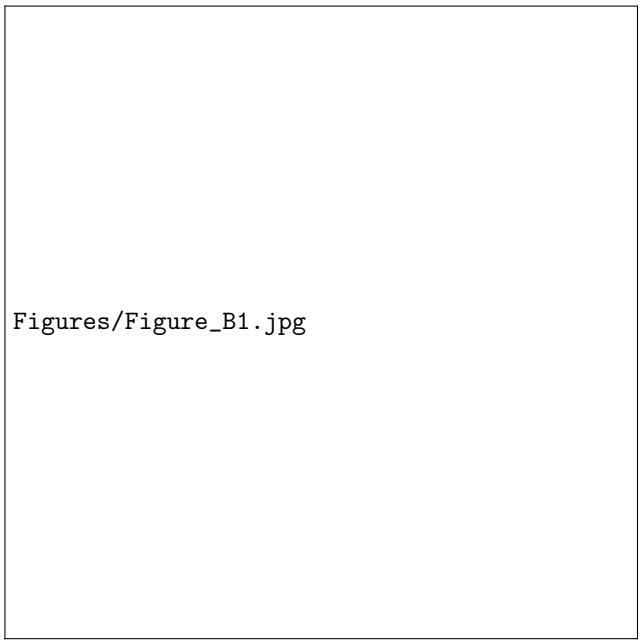


Figure B1. Distribution of magnitudes in chunk of INSTANCE data used for testing. Once again, no resampling is applied to the dataset based on magnitude.

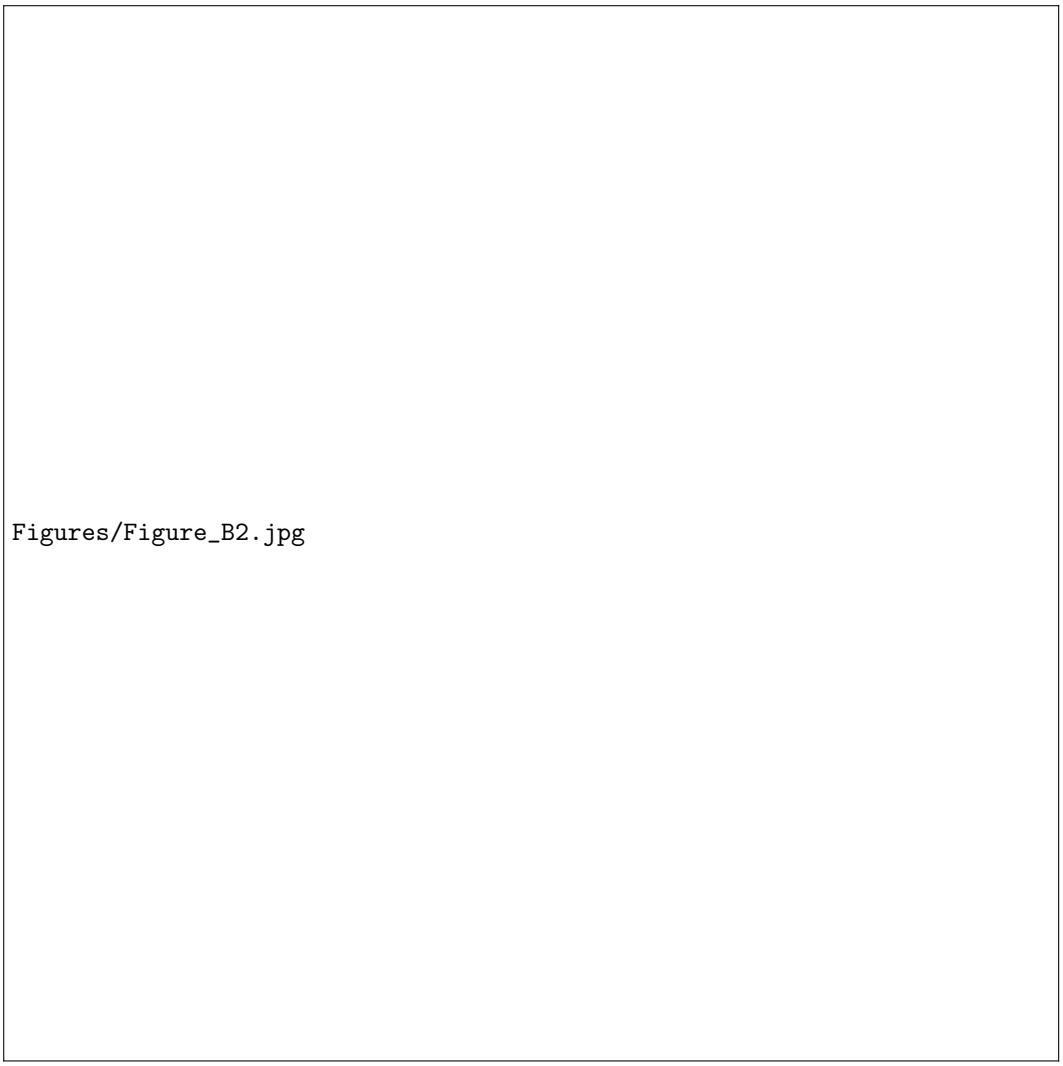


Figure B2. Analysis of model performance as a regressor on INSTANCE Data, here the events misclassified as noise, reflect the imbalanced distribution of magnitudes in the dataset itself, whereas the predicted magnitude of noise waveforms follows a similar trend as in case of STEAD data. (a) Confusion matrix for classifier performance on Instance Data. (b) Distribution of predicted magnitudes of noise misclassified as event and true magnitude of events misclassified as noise.

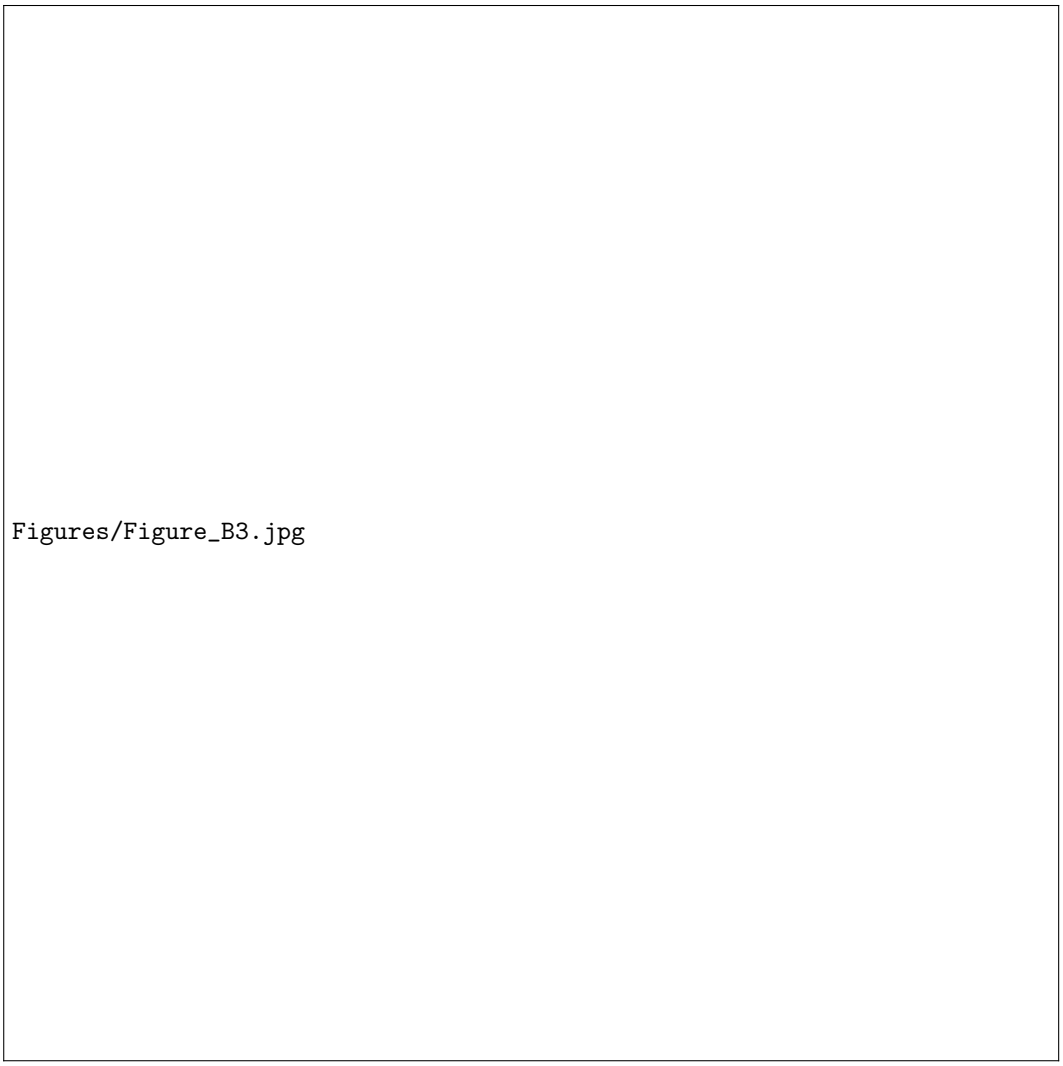


Figure B3. Analysis of model performance as a regressor on INSTANCE Data. (a) Relationship between true and predicted magnitude values. (b) Distribution of errors in predicted magnitudes. (c) Distribution of errors in P-arrival estimation.

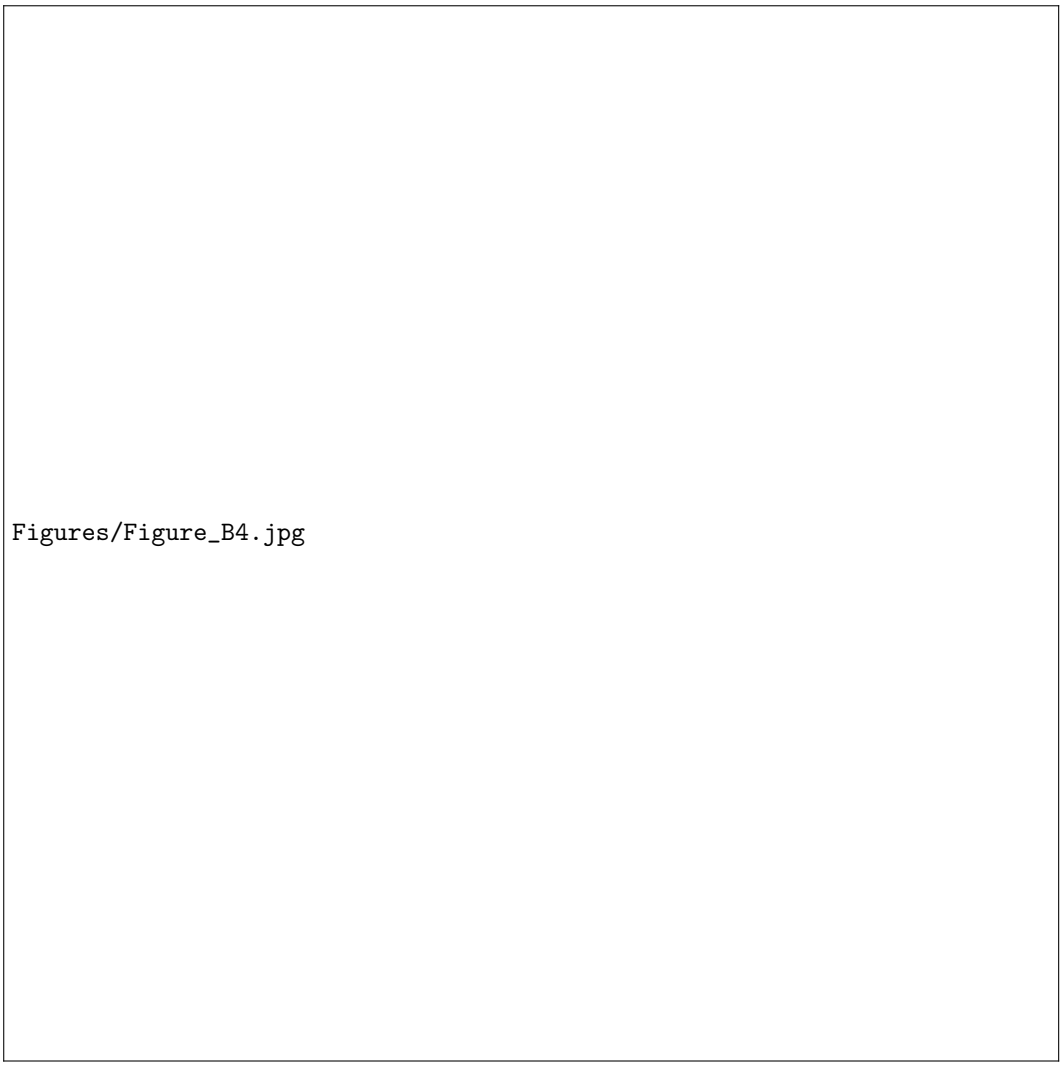


Figure B4. (a) Variation of errors with signal to noise ratio. (b) Variation of errors with hypocentral distance.

Appendix C Effect of using different types of ground motion data as input

We compared the performance of the model when trained on different kinds of ground motion data viz. acceleration (in μm s^{-2}), velocity (in nm s^{-1}) and displacement (in nm) to investigate the effects of instrument response removal. This part of the analysis was done only on the STEAD data. We lose roughly one fourth of the data due to unavailability of the instrument response. In each case, the models were trained on roughly the same number of traces, alongside which we also compare it with the model whose results are discussed in the Results (trained on more traces). For a fair comparison, we also train a model on raw data, using roughly the same number of traces as in case of ground motion data, accounting for the loss of data due to unavailability of instrument response (this model is referred to as raw data (smaller) in the tables). The reason behind doing this is to highlight one of the advantages of using counts data without instrument response removal, which is the availability of more traces for training and testing. All five models have been tested on the same traces.

Tables C1-C3 show the comparison between different types of input data. Even though certain ground motion parameters perform better in some metrics, using the raw data gives us an advantage that the data can be used in real time, and it is much readily available.

Table C1. Summary of classification performance for different types of ground motion data; removing instrument response does not seem to provide a significant advantage over using raw data

	Metric						
		Precision (%)		**Recall (%)**		**F1-score (%)**	
Type of input data	**Accuracy (%)**	Event	Noise	Event	Noise	Event	Noise
Raw Data	**98.33**	99.90	85.90	**98.25**	99.09	**99.06**	92.03
Raw Data (smaller)	97.85	**99.91**	82.36	97.71	99.17	98.79	89.99
Acceleration	98.15	99.79	**93.62**	97.75	**99.37**	98.76	**96.41**
Velocity	97.81	99.65	92.74	97.42	98.98	98.53	95.76
Displacement	96.52	99.61	88.54	95.74	98.86	97.64	93.41

Table C2. Summary of magnitude estimation for different types of ground motion data

	Metric			
Ground Motion	**Mean Error**	**St. dev. of Error**	**RMSE**	**MAE**
Raw Data	-0.19	0.63	0.65	0.50
Raw Data (smaller)	**0.01**	0.64	0.64	0.49
Acceleration	-0.11	**0.56**	**0.57**	**0.44**
Velocity	-0.09	0.62	0.63	0.47
Displacement	-0.32	0.65	0.72	0.54

Table C3. Summary of P-arrival estimation for different types of ground motion data

	Metric			
Ground Motion	**Mean Error (s)**	**St. dev. of Error (s)**	**RMSE (s)**	**MAE (s)**
Raw Data	**-0.04**	**0.11**	**0.12**	**0.08**
Raw Data (smaller)	-0.06	**0.11**	0.13	0.09
Acceleration	-0.07	0.12	0.14	0.10
Velocity	-0.06	0.14	0.15	0.11
Displacement	-0.06	0.18	0.19	0.13

Acknowledgments

This research is supported by the "KI-Nachwuchswissenschaftlerinnen" - grant SAI 01IS20059 by the Bundesministerium für Bildung und Forschung - BMBF. Calculations were performed at the Frankfurt Institute for Advanced Studies' new GPU cluster, funded by BMBF for the project Seismologie und Artifizielle Intelligenz (SAI). The authors are also thankful to Dr. KiranKumar Thingbaijam, Dr. Jan Steinheimer and Jonas Köhler for their kind suggestions. H.St. gratefully acknowledges the Judah M. Eisenberg Laureatus - Professur at Fachbereich Physik, Goethe Universität Frankfurt, funded by the Walter Greiner Gesellschaft zur Förderung der physikalischen Grundlagenforschung e.V.

Open Research

The seismic waveforms used in our research are a part of two datasets – STanford EArthquake Dataset (STEAD) (Mousavi et al., 2019) which was downloaded `https://github.com/smousavi05/STEAD` (last accessed January 2022) and INSTANCE (Michelini, Cianetti, Gaviano, Giunchi, Jozinović, & Lauciani, 2021) which was downloaded from `https://doi.org/10.13127/instance` (last accessed January 2022).

References

Allen, R., Gasparini, P., Kamigaichi, O., & Böse, M. (2009). The status of earthquake early warning around the world: An introductory overview. *Seismol. Res. Lett.*, *80*, 682–693. doi: 10.1785/gssrl.80.5.682

Allen, R., & Kanamori, H. (2003, 06). The potential for earthquake early warning in southern california. *Science (New York, N.Y.)*, *300*, 786-9. doi: 10.1126/science.1080912

Allen, R. M., & Melgar, D. (2019). Earthquake early warning: Advances, scientific challenges, and societal needs. *Annu Rev Earth Planet Sci*, *47*(1), 361-388. Retrieved from `https://doi.org/10.1146/annurev-earth-053018-060457` doi: 10.1146/annurev-earth-053018-060457

Allen, R. V. (1978). Automatic earthquake recognition and timing from single traces. *Bull. Seismol. Soc. Am.*, *68*(5), 1521–1532. Retrieved from `https://doi.org/10.1785/BSSA0680051521`

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., & Wassermann, J. (2010, 06). Obspy: A python toolbox for seismology. *Seismol. Res. Lett.*, *81*, 530-533. doi: 10.1785/gssrl.81.3.530

Chakraborty, M., Li, W., Faber, J., Rümpker, G., Stöcker, H., & Srivastava, N. (2021). *A study on the effect of input data length on deep learning based magnitude classifier.*

Charles, P. (2013). *Project title.* GitHub. Retrieved from `https://github.com/charlespwd/project-title`

Chollet, F. (2017). *Deep learning with python.* Manning.

Chung, D. H., & Bernreuter, D. L. (1981). Regional relationships among earthquake

magnitude scales. *Rev. Geophys.*, *19*(4), 649-663. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/RG019i004p00649 doi: https://doi.org/10.1029/RG019i004p00649

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, *12*(ARTICLE), 2493–2537. doi: 10.5555/1953048.2078186

Cremen, G., & Galasso, C. (2020). Earthquake early warning: Recent advances and perspectives. *Earth Sci Rev*, *205*, 103184. doi: https://doi.org/10.1016/j.earscirev.2020.103184

Ekström, G., & Dziewonski, A. (1988). Evidence of bias in estimations of earthquake size. *Nature*, *332*, 319–323. Retrieved from https://doi.org/10.1038/332319a0

Fenner, D., Rümpker, G., Li, W., Chakraborty, M., Faber, J., Köhler, J., ... Srivastava, N. (2022). Automated seismo-volcanic event detection applied to stromboli (italy). *Frontiers in Earth Science*, *10*. Retrieved from https://www.frontiersin.org/article/10.3389/feart.2022.809037 doi: 10.3389/feart.2022.809037

Geller, R. (1976). Scaling relations for earthquake source parameters and magnitudes. *Bull. Seismol. Soc. Am.*, *66(5)*, 1501-1523. Retrieved from https://doi.org/10.1785/BSSA0660051501

Giardini, D. (1988). Frequency distribution and quantification of deep earthquakes. *J. Geophys. Res. Solid Earth*, *93(B3)*, 2095-2105. doi: 10.1029/JB093iB03p02095

Gutenberg, B., & Richter, C. (1944). Frequency of earthquakes in california. *Bull. Seismol. Soc. Am.*, *34*, 185-188.

Hara, S., Fukahata, Y., & Iio, Y. (2019). P-wave first-motion polarity determination of waveform data in western Japan using deep learning. *Earth Planets Space*, *71*(127). doi: https://doi.org/10.1186/s40623-019-1111-x

Harris, C. R., Millman, K. J., van der Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357-362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

He, K., Ren, S., Sun, J., & Zhang, X. (2016). Deep residual learning for image recognition. *IEEE Conf Comput Vis Pattern Recognit(CVPR)*, 770-778. doi: 10.1109/CVPR.2016.90

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., ... Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.*, *29*(6), 82-97. doi: 10.1109/MSP.2012.2205597

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Howell Jr, B. (1981). On the saturation of earthquake magnitudes. *Bull. Seismol. Soc. Am.*, *71(5)*, 1401-1422. Retrieved from https://doi.org/10.1785/BSSA0710051401

Jin, X., Zhang, H., Li, J., Wei, Y., & Ma, Q. (2013). Earthquake magnitude estimation using the τ_c and p_d method for earthquake early warning systems. *Earthq. Sci.*, *26*(es-26-1-23), 23-31. doi: 10.1007/s11589-013-0005-4

Kanamori, H. (1983). Magnitude scale and quantification of earthquakes. *Tectonophysics*, *93(3-4)*, 185-199. Retrieved from https://doi.org/10.1016/0040-1951(83)90273-1

Kanamori, H. (2005). Real-time seismology and earthquake damage mitigation. *Annu Rev Earth Planet Sci*, *33*(1), 195-214. doi: 10.1146/annurev.earth.33.092203.122626

Kanamori, H., & Stewart, G. S. (1978). Seismological aspects of the guatemala earthquake of february 4, 1976. *J. Geophys. Res. Solid Earth*, *83*(B7), 3427-

3434. doi: https://doi.org/10.1029/JB083iB07p03427

Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015). Convolutional neural networks for patient-specific ecg classification. , 2608-2611. doi: 10.1109/EMBC.2015.7318926

Kong, Q., Allen, R. M., & Schreier, L. (2016). Myshake: Initial observations from a global smartphone seismic network. *Geophys. Res. Lett.*, *43*, 9588–9594. doi: 10.1002/2016GL070955.

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2018). Machine learning in seismology: Turning data into insights. *Seismol. Res. Lett.*, *90*(1), 3-14. Retrieved from https://doi.org/10.1785/0220180259

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Prog. Artif. Intell.*, *5*, 221–232. Retrieved from https://doi.org/10.1007/s13748-016-0094-0

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, *60*(6), 84–90. Retrieved from https://doi.org/10.1145/3065386

Kuyuk, H. S., & Allen, R. M. (2013). A global approach to provide magnitude estimates for earthquake early warning alerts. *Geophys. Res. Lett.*, *40*(24), 6329-6333. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2013GL058580 doi: https://doi.org/10.1002/2013GL058580

Kuyuk, H. S., & Susumu, O. (2018). Real-time classification of earthquake using deep learning. *Proc. Comput. Sci*, *140*, 298–305. Retrieved from https://doi.org/10.1016/j.procs.2018.10.316

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. Retrieved from https://doi.org/10.1038/nature14539

Li, W., Chakraborty, M., Fenner, D., Faber, J., Zhou, K., Rümpker, G., ... Srivastava, N. (2021). *Epick: Multi-class attention-based u-shaped neural network for earthquake detection and seismic phase picking.* Retrieved from https://arxiv.org/abs/2109.02567

Li, W., Sha, Y., Zhou, K., Faber, J., Rümpker, G., Stöcker, H., & Srivastava, N. (2022). *Deep learning-based small magnitude earthquake detection and seismic phase classification.* Retrieved from https://arxiv.org/abs/2204.02870

Li, Z., Meier, M. A., Hauksson, E., Zhan, Z., & Andrews, J. (2018). Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophys. Res. Lett.*, *45*(10), 4773-4779. Retrieved from https://doi.org/10.1029/2018GL077870

Liao, W., Lee, E., Mu, D., Chen, P., & Rau, R. (2021, 03). ARRU Phase Picker: Attention Recurrent-Residual U-Net for Picking Seismic P- and S-Phase Arrivals. *Seismol. Res. Lett.*, *92*(4), 2410-2428. Retrieved from https://doi.org/10.1785/0220200382 doi: 10.1785/0220200382

Lomax, A., Michelini, A., & Jozinović, D. (2019, 02). An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. *Seismol. Res. Lett.*, *90*, 517-529. doi: 10.1785/0220180311

Meier, M., Ross, Z. E., Ramachandran, A., Balakrishna, A., Nair, S., Kundzicz, P., ... Yue, Y. (2019). Reliable real-time seismic signal/noise discrimination with machine learning. *J. Geophys. Res. Solid Earth*, *124*(1), 788-800. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JB016661 doi: https://doi.org/10.1029/2018JB016661

Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinovic, D., & Lauciani, V. (2021). Instance – the italian seismic dataset for machine learning. *Earth Syst. Sci. Data Discuss.*, *2021*, 1–47. doi: 10.5194/essd-2021-164

Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinović, D., & Lauciani, V. (2021). Instance the italian seismic dataset for machine learning, seismic waveforms and associated metadata. *Istituto Nazionale di Geofisica e Vulcanologia*

(INGV). Retrieved from https://doi.org/10.13127/instance

Mikolov, T., Deoras, A., Povey, D., Burget, L., & Černocký, J. (2011). Strategies for training large scale neural network language models. In *2011 ieee workshop on automatic speech recognition understanding* (p. 196-201). doi: 10.1109/ASRU.2011.6163930

Mousavi, S. M., & Beroza, G. C. (2020). A machine-learning approach for earthquake magnitude estimation. *Geophys. Res. Lett.*, *47*, *e2019GL085976.*. Retrieved from https://doi.org/10.1029/2019GL085976

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L., & Beroza, G. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.*, *11*(3952). doi: https://doi.org/10.1038/s41467-020-17591-w

Mousavi, S. M., Sheng, Y., Zhu, W., & Beroza, G. C. (2019). Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, *7*, 179464-179476. Retrieved from https://doi.org/10.1109/ACCESS.2019.2947848

Münchmeyer, J., Bindi, D., Leser, U., & Tilmann, F. (2020). The transformer earthquake alerting model: a new versatile approach to earthquake early warning. *Geophys. J. Int.*, *225*(1), 646-656. Retrieved from https://doi.org/10.1093/gji/ggaa609 doi: 10.1093/gji/ggaa609

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., ... Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 ieee international conference on signal and image processing applications (icsipa)* (p. 342-347). doi: 10.1109/ICSIPA.2011.6144164

Nakamura, Y. (1988). On the urgent earthquake detection and alarm system (uredas). *9th world conference on earthquake engineering*, *VII*(B7), 673-678.

Panakkat, A., & Adeli, H. (2009). Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Comput.-Aided Civ. Infrastruct. Eng.*, *24*(4), 280-292. Retrieved from https://doi.org/10.1111/j.1467-8667.2009.00595.x

Perol, T., Gharbi, M., & Denolle, M. (2018). Convolutional neural network for earthquake detection and location. *Sci. Adv.*, *4*(2), e1700578. doi: 10.1126/sciadv.1700578

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep contextualized word representations. , 2227–2237. Retrieved from https://aclanthology.org/N18-1202 doi: 10.18653/v1/N18-1202

Prechelt, L. (2012). Early stopping — but when? In *Neural networks: Tricks of the trade: Second edition* (pp. 53–67). Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-35289-8_5 doi: 10.1007/978-3-642-35289-8_5

Richter, C. F. (1935). An instrumental earthquake magnitude scale. *Bull. Seismol. Soc. Am.*, *25*, 1-32. Retrieved from https://doi.org/10.1785/BSSA0250010001

Ross, Z. E., Meier, M., & Hauksson, E. (2018). P wave arrival picking and first-motion polarity determination with deep learning. *J. Geophys. Res. Solid Earth*, *123*(6), 5120-5129. Retrieved from https://doi.org/10.1029/2017JB015251

Tieleman, T., & Hinton, G. (2012). *Lecture 6.5-rmsprop, coursera: Neural networks for machine learning.* University of Toronto, Technical Report.

Ting, K. M. (2017). Confusion matrix. In *Encyclopedia of machine learning and data mining* (pp. 260–260). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-1-4899-7687-1_50 doi: 10.1007/978-1-4899-7687-1_50

Wu, Y., & Zhao, L. (2006). Magnitude estimation using the first three seconds p-wave amplitude in earthquake early warning. *Geophys. Res. Lett.*, *33*(16). Retrieved from https://doi.org/10.1029/2006GL026871

Yanwei, W., Xiaojun, L., Zifa, W., Jianping, S., & Enhe, B. (2021). Deep learning for P-wave arrival picking in earthquake early warning. *Earthq. Eng. Eng. Vib.*, *20*, 391–402. doi: 10.1007/s11803-021-2027-6

Zhou, Y., Yue, H., Kong, Q., & Zhou, S. (2019, 04). Hybrid Event Detection and Phase-Picking Algorithm Using Convolutional and Recurrent Neural Networks. *Seismological Research Letters*, *90*(3), 1079-1087. Retrieved from https://doi.org/10.1785/0220180319 doi: 10.1785/0220180319

Zhu, J., Li, S., Song, J., & Wang, Y. (2021). Magnitude estimation for earthquake early warning using a deep convolutional neural network. *Frontiers in Earth Science*, *9*, 341. doi: 10.3389/feart.2021.653226

Zhu, W., & Beroza, G. C. (2019). Phasenet: a deep-neural-network-based seismic arrival-time picking method. *Geophys. J. Int.*, *216*(1), 261-273. Retrieved from https://doi.org/10.1093/gji/ggy423

Ziv, A. (2014). New frequency-based real-time magnitude proxy for earthquake early warning. *Geophys. Res. Lett.*, *41*(16), 7035– 7040. Retrieved from https://doi.org/10.1002/2014GL061564

Figure1.

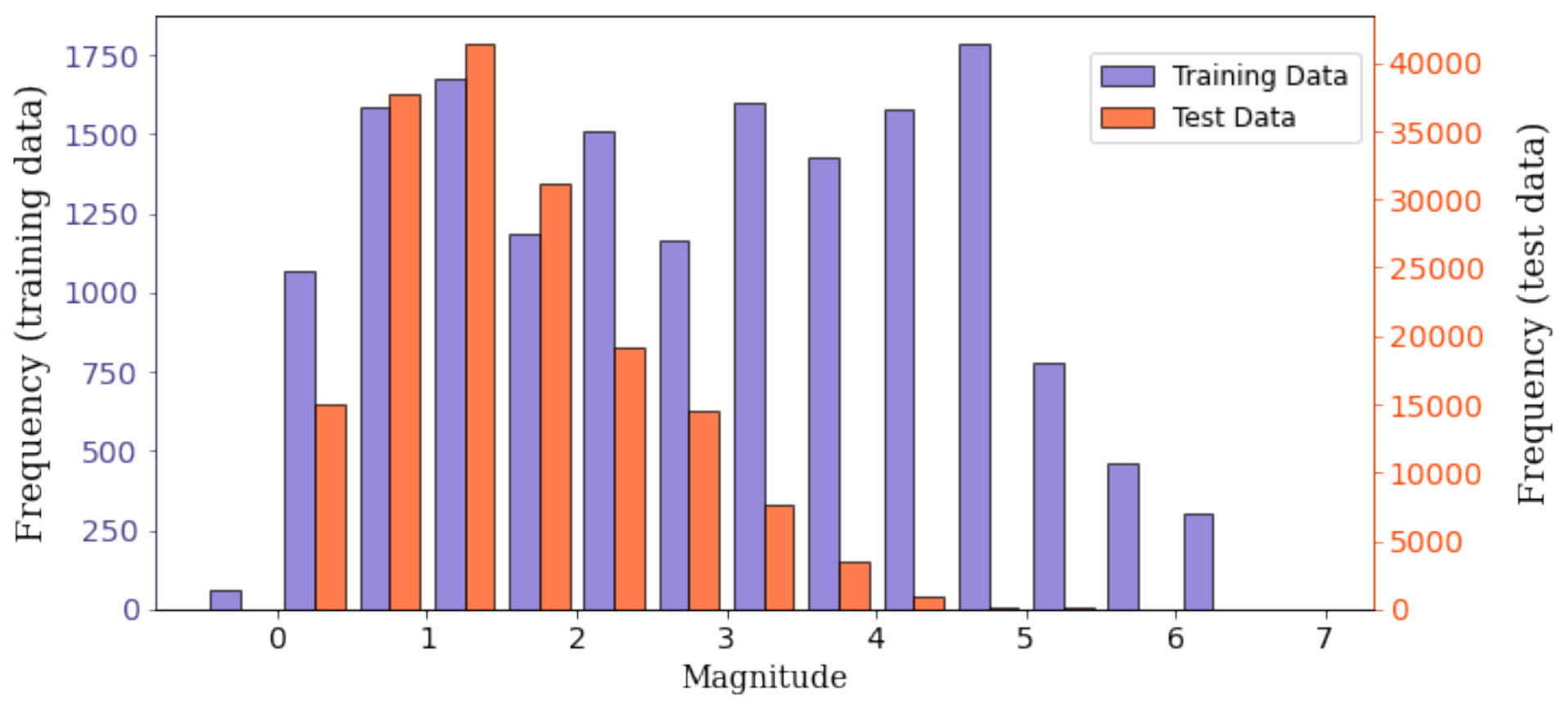
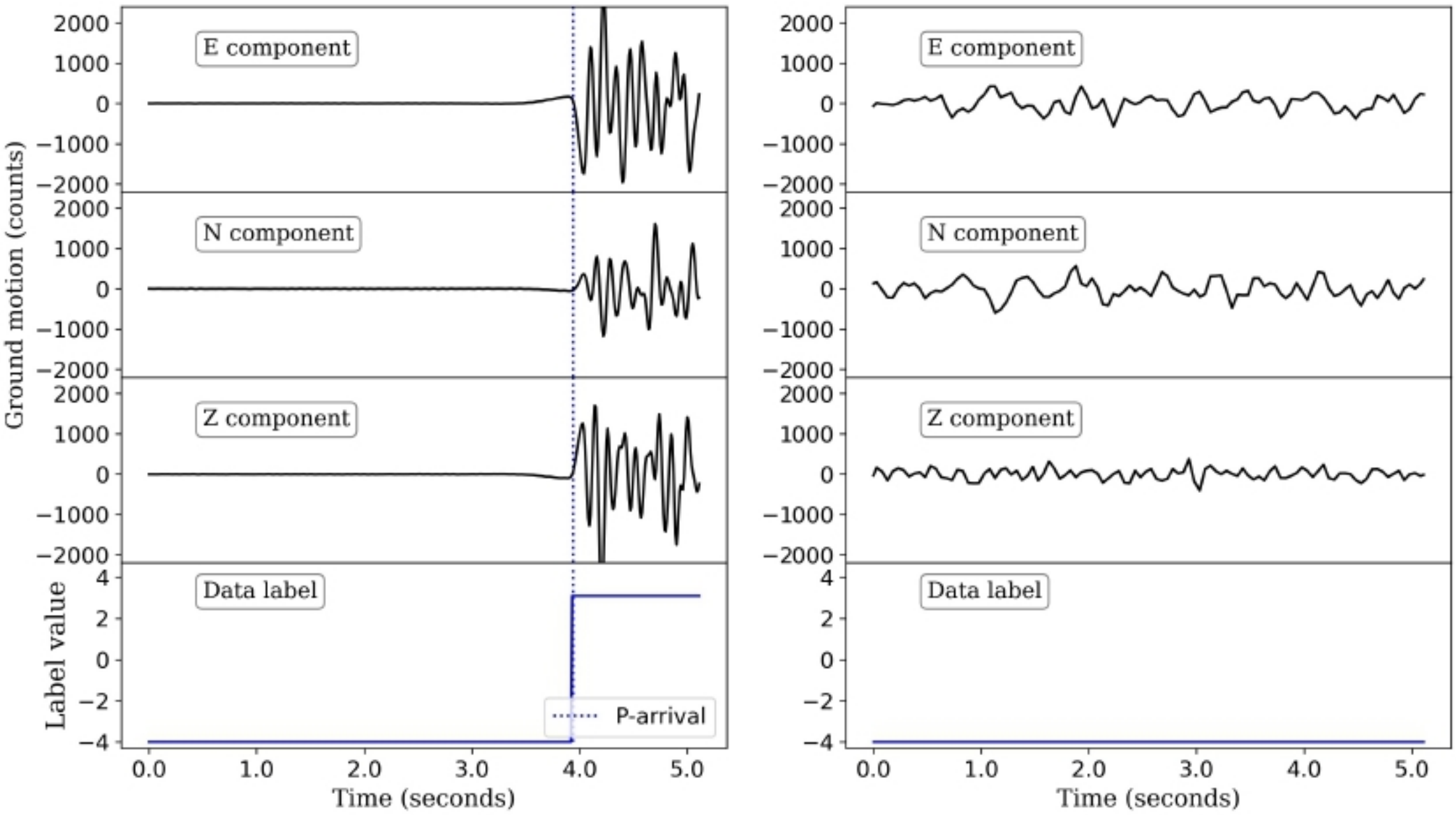

Training Data
Test Data
Frequency (training data)
Frequency (test data)
Magnitude
0
250
500
750
1000
1250
1500
1750
0
5000
10000
15000
20000
25000
30000
35000
40000
0
1
2
3
4
5
6
7

Figure2.

(a)


E component
N component
Z component
Data label
P-arrival
Ground motion (counts)
Label value
Time (seconds)
2000
1000
0
−1000
−2000
4
2
0
−2
−4
0.0
1.0
2.0
3.0
4.0
5.0

(b)

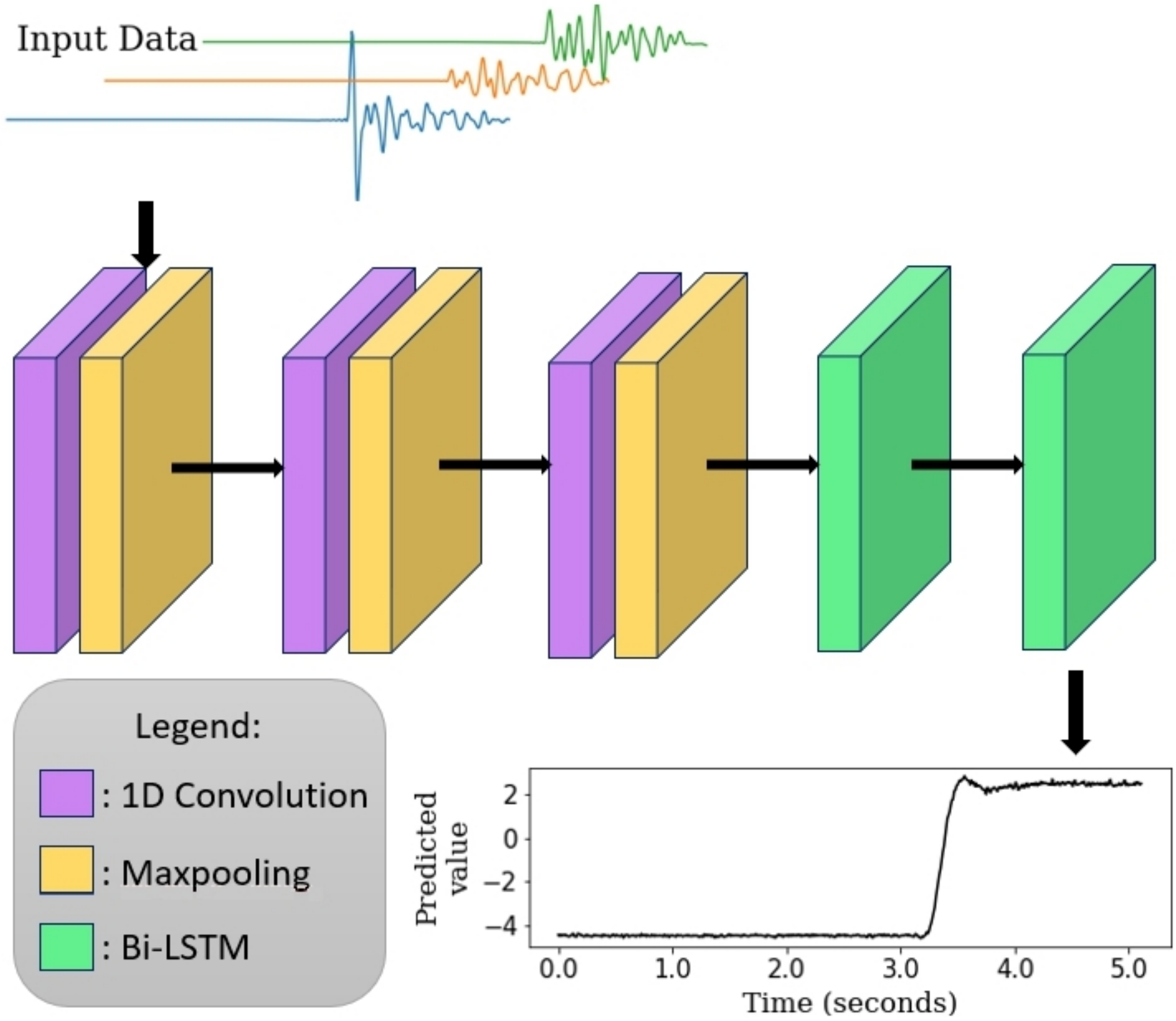

Input Data
Legend:
: 1D Convolution
: Maxpooling
: Bi-LSTM
Predicted value
Time (seconds)
2
0
−2
−4
0.0
1.0
2.0
3.0
4.0
5.0

Figure3.

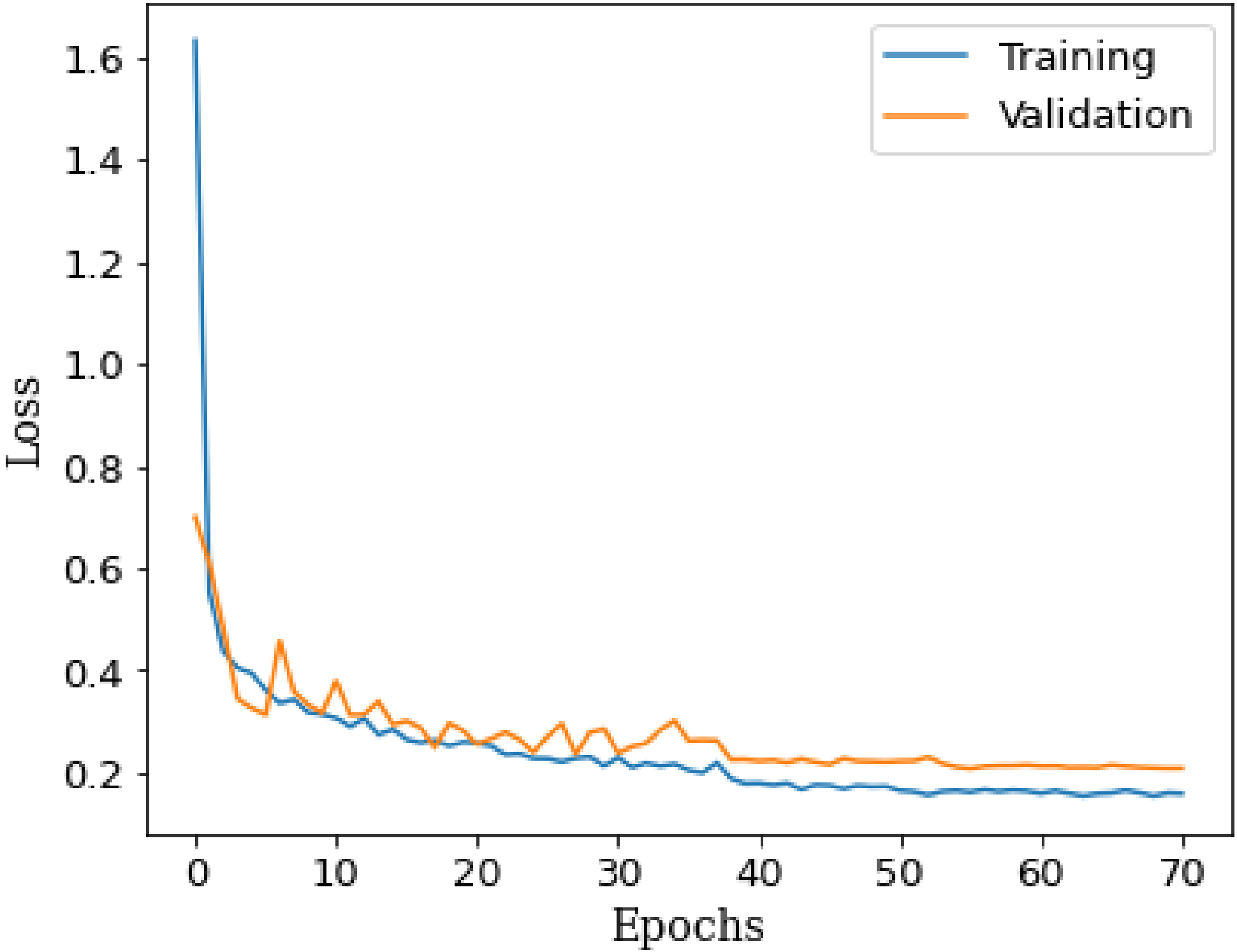
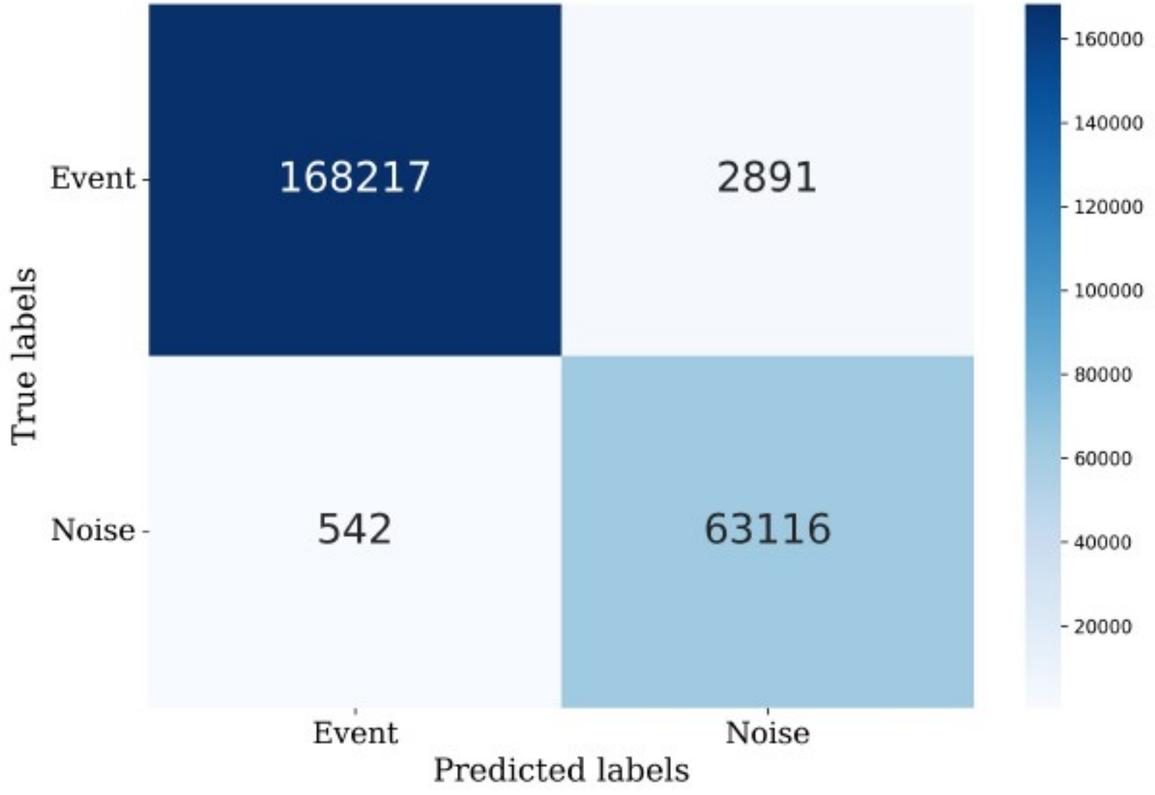

Training
Validation
Loss
Epochs
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50
60
70

Figure4.

(a)


168217
2891
542
63116
Event
Noise
True labels
Event
Noise
Predicted labels
160000
140000
120000
100000
80000
60000
40000
20000

(b)

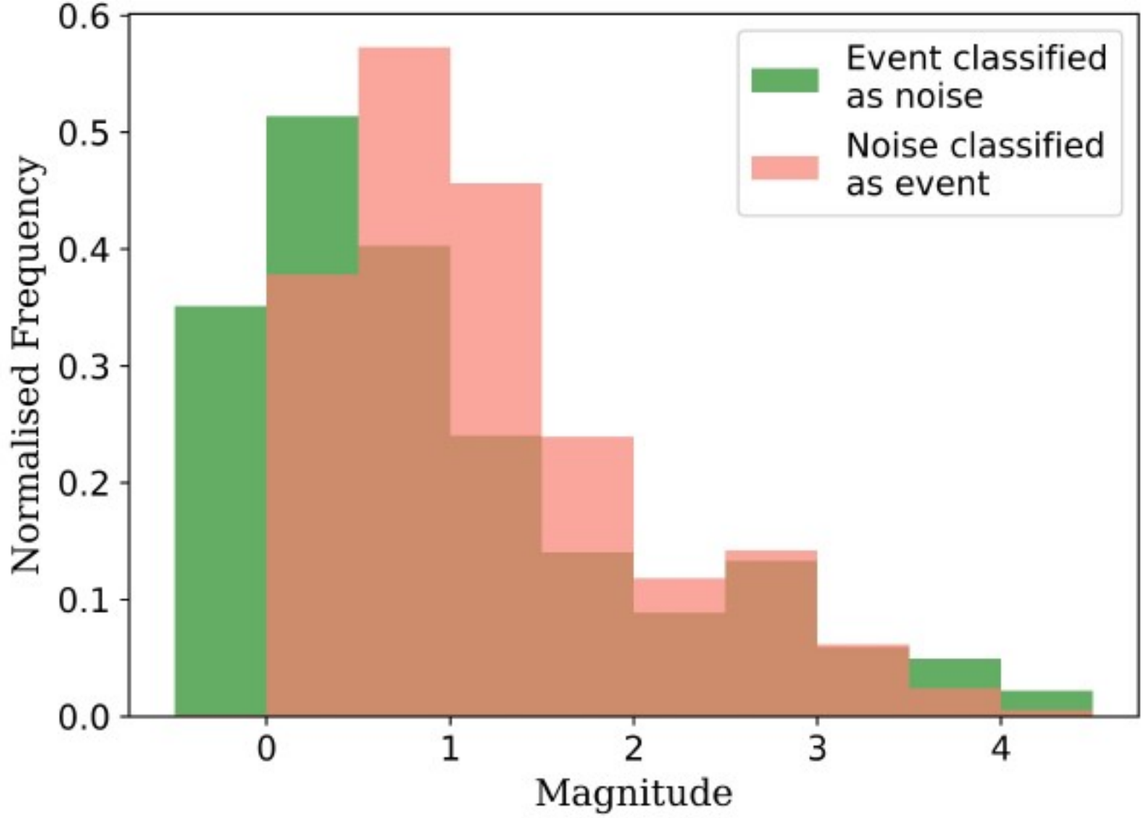
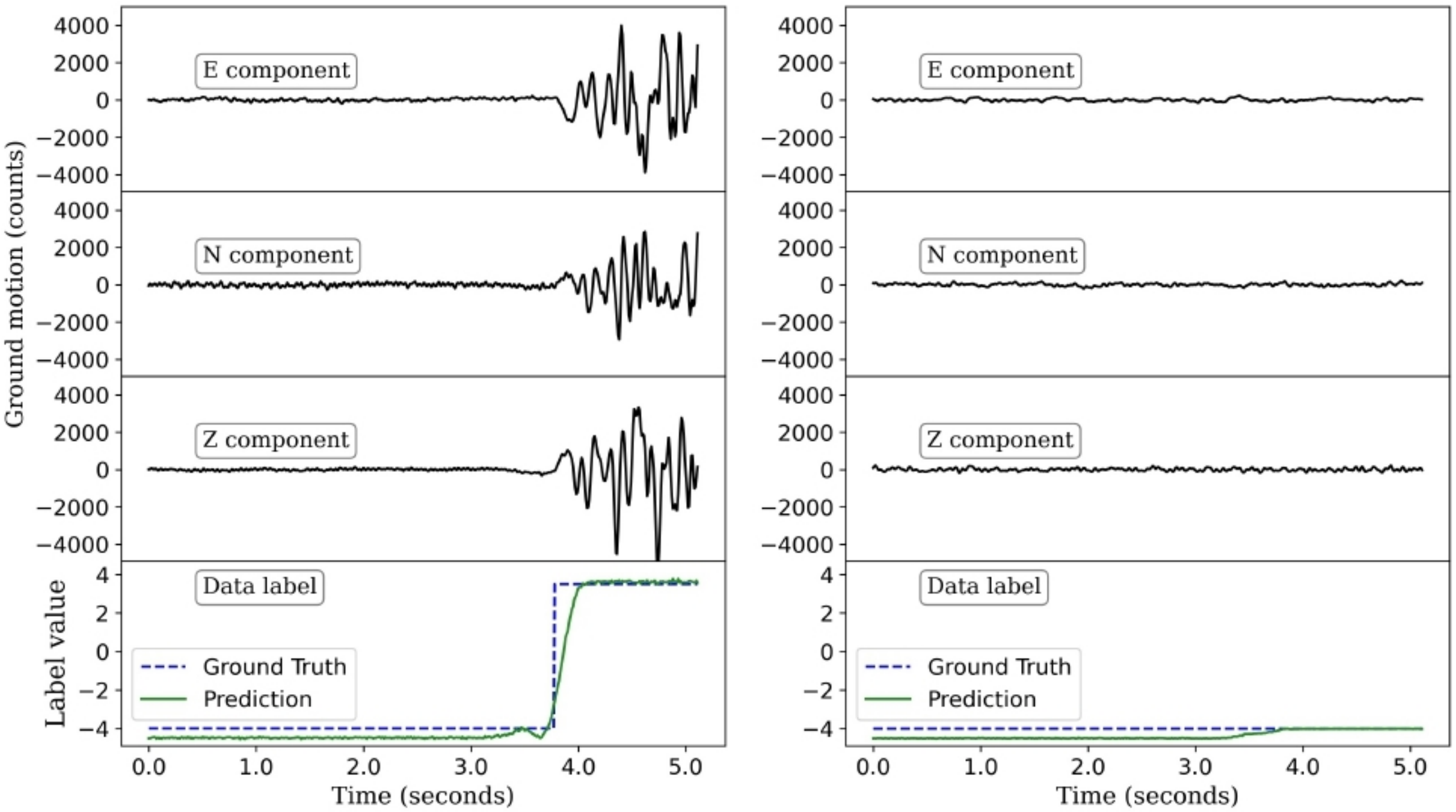

Event classified as noise
Noise classified as event
Normalised Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
Magnitude

Figure5.

(a)


E component
N component
Z component
Data label
Ground Truth
Prediction
Ground motion (counts)
Label value
Time (seconds)

(b)

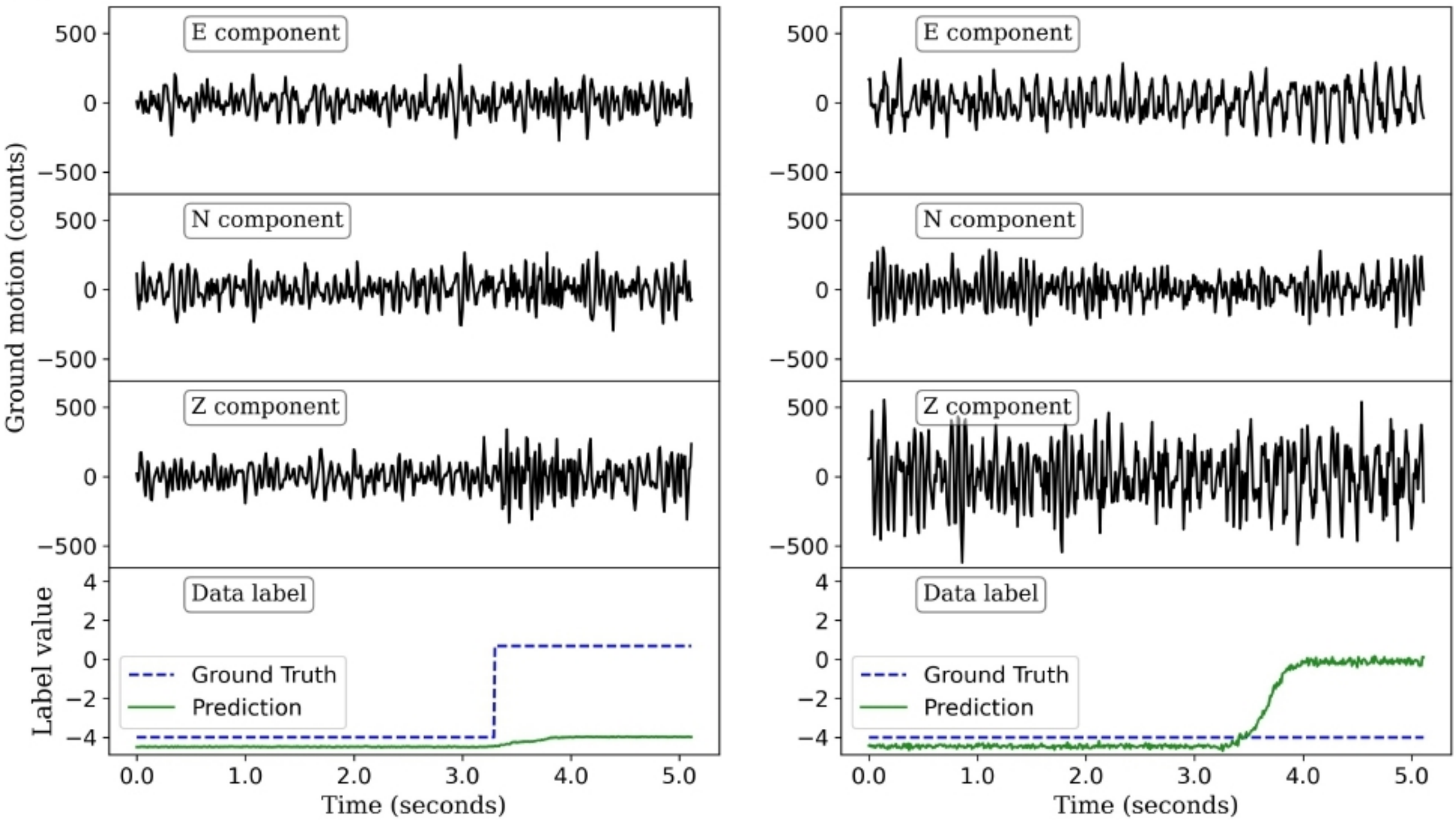
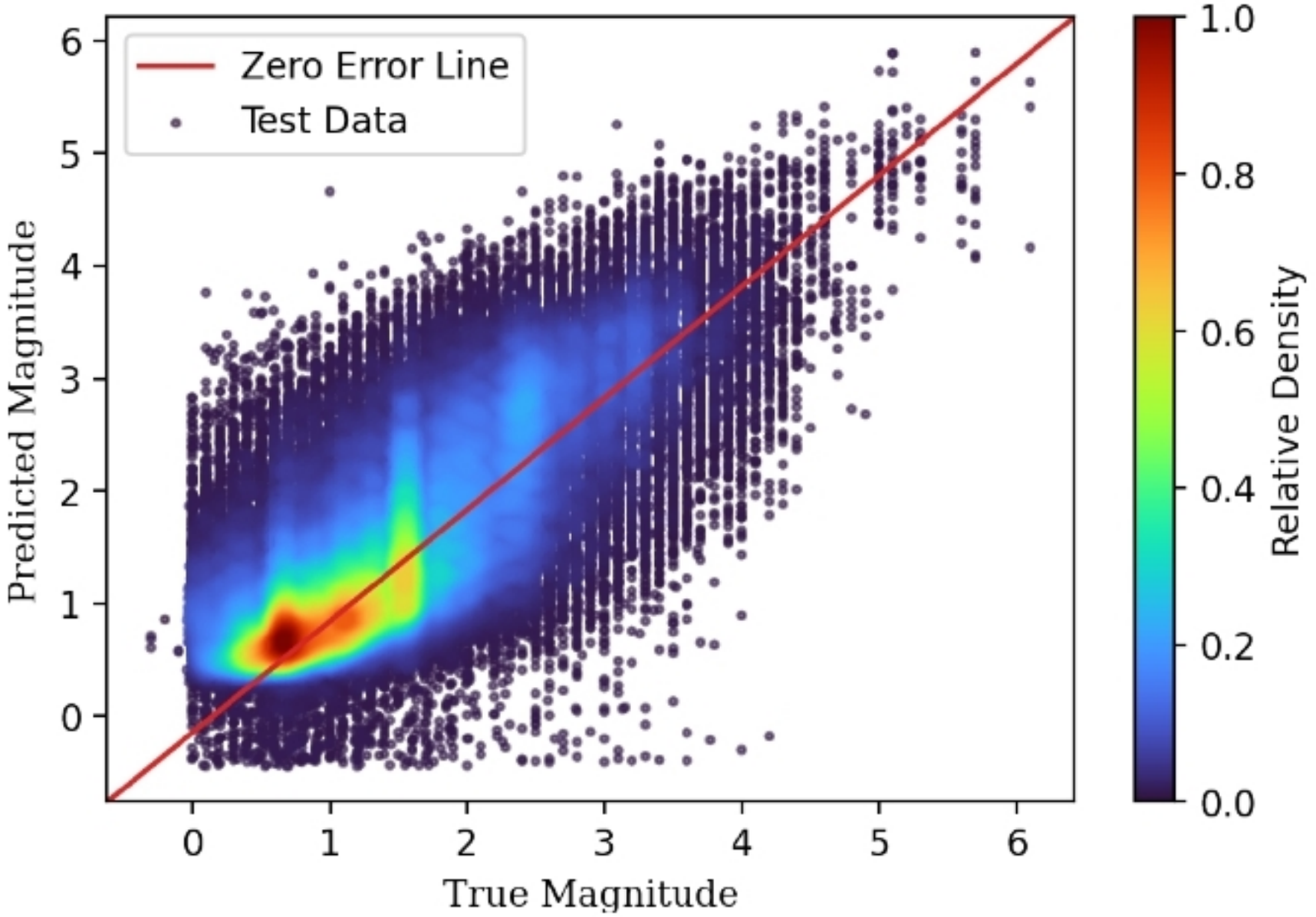
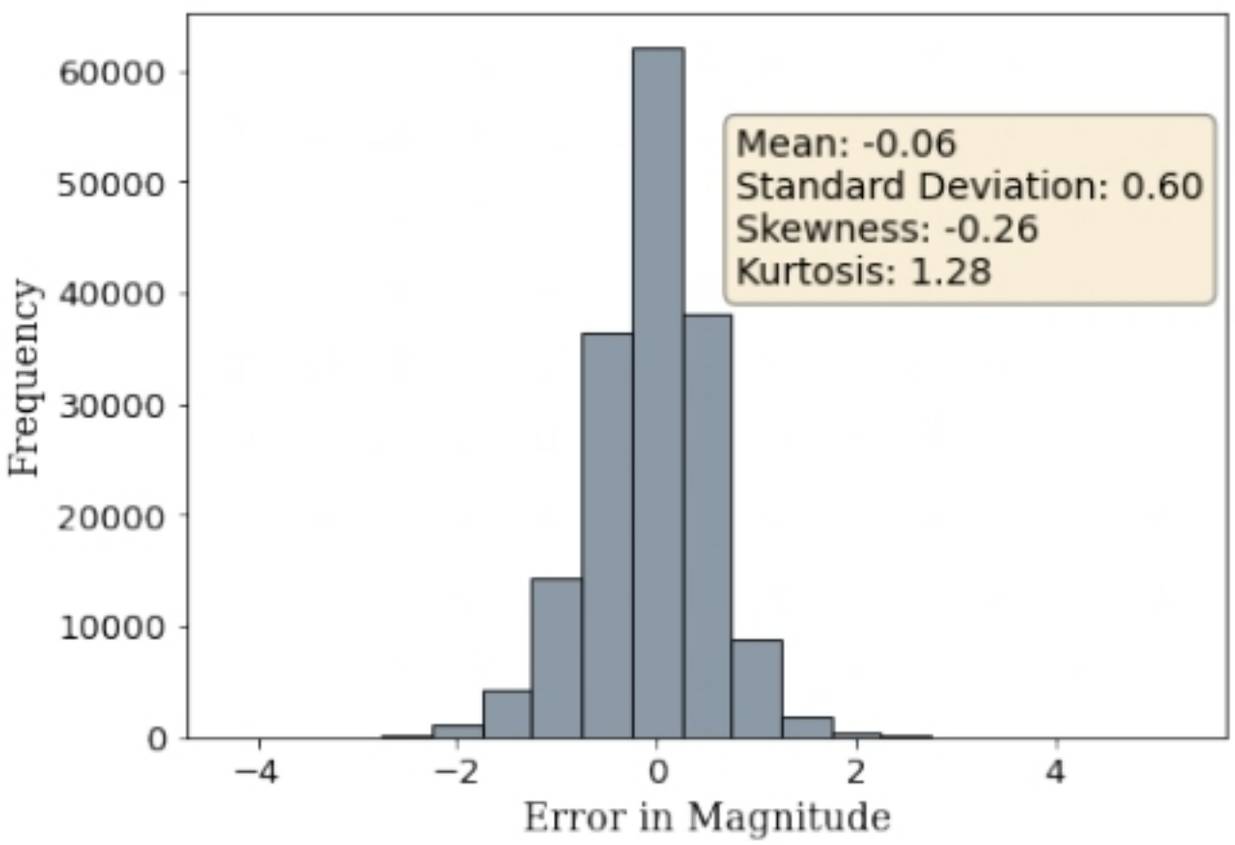

E component
N component
Z component
Data label
Ground Truth
Prediction
Ground motion (counts)
Label value
Time (seconds)

Figure6.

(a)


Zero Error Line
Test Data
Predicted Magnitude
True Magnitude
Relative Density
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0

(b)


Mean: -0.06
Standard Deviation: 0.60
Skewness: -0.26
Kurtosis: 1.28
Frequency
Error in Magnitude
0
10000
20000
30000
40000
50000
60000
−4
−2
0
2
4

(c)

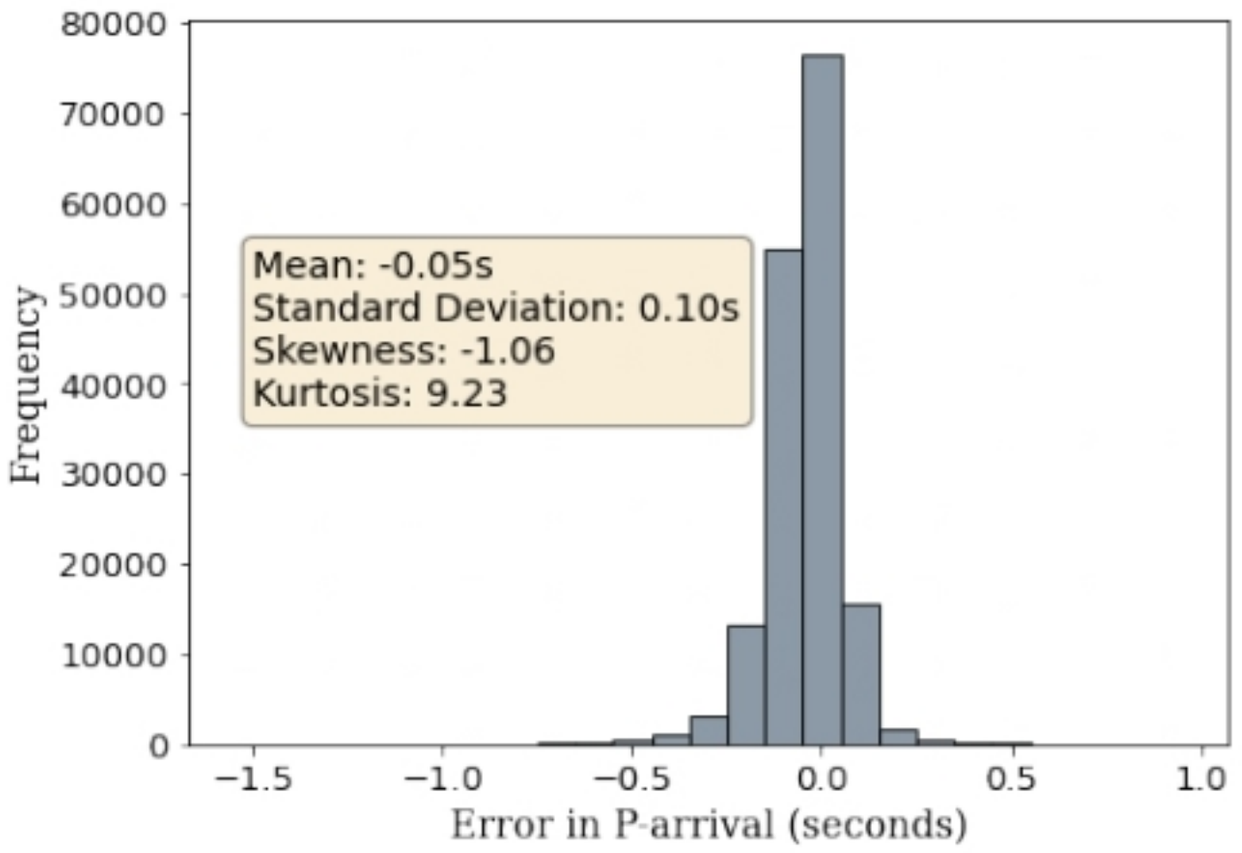
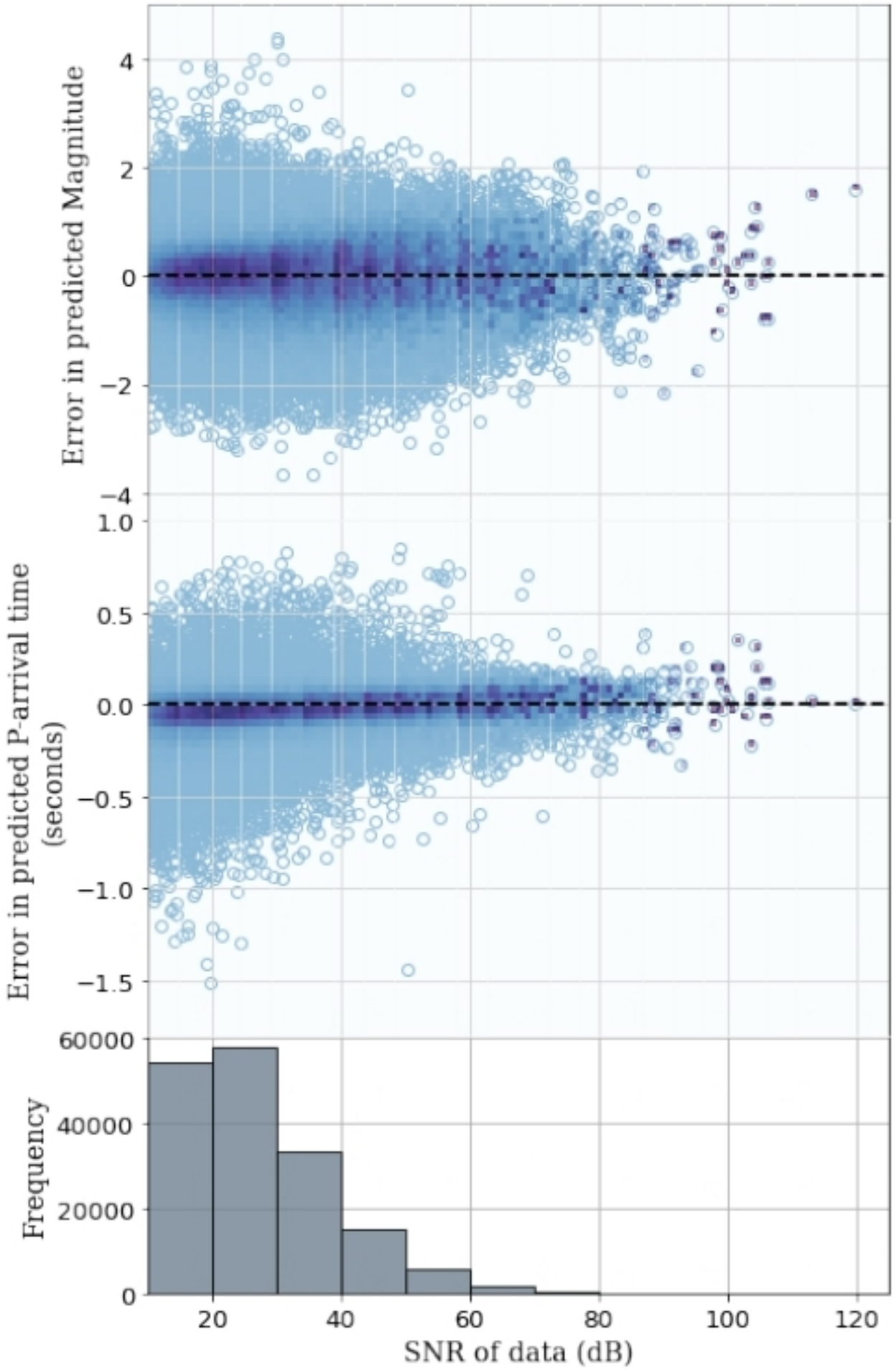
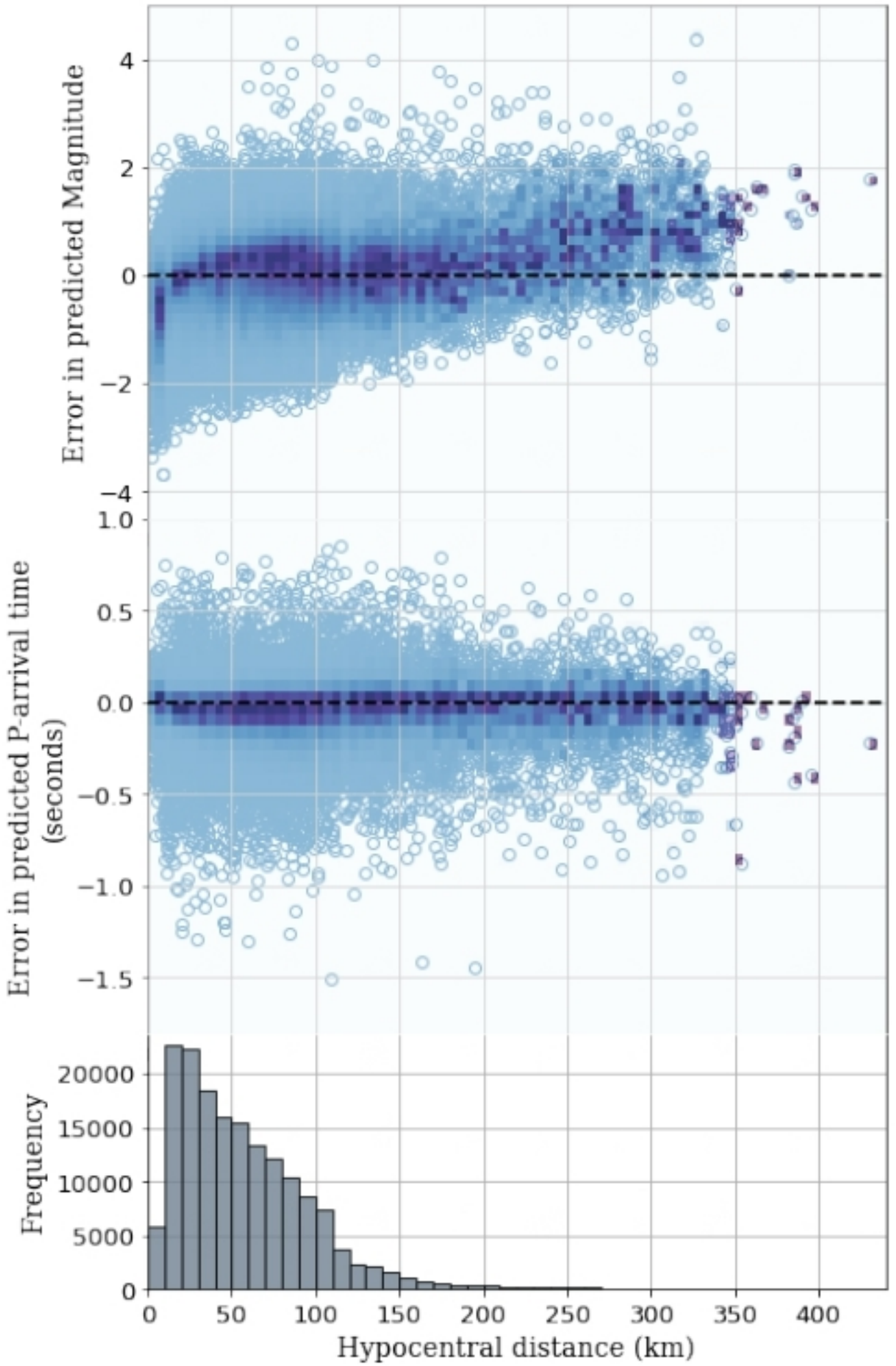

Mean: -0.05s
Standard Deviation: 0.10s
Skewness: -1.06
Kurtosis: 9.23
Frequency
Error in P-arrival (seconds)
0
10000
20000
30000
40000
50000
60000
70000
80000
−1.5
−1.0
−0.5
0.0
0.5
1.0

Figure7.

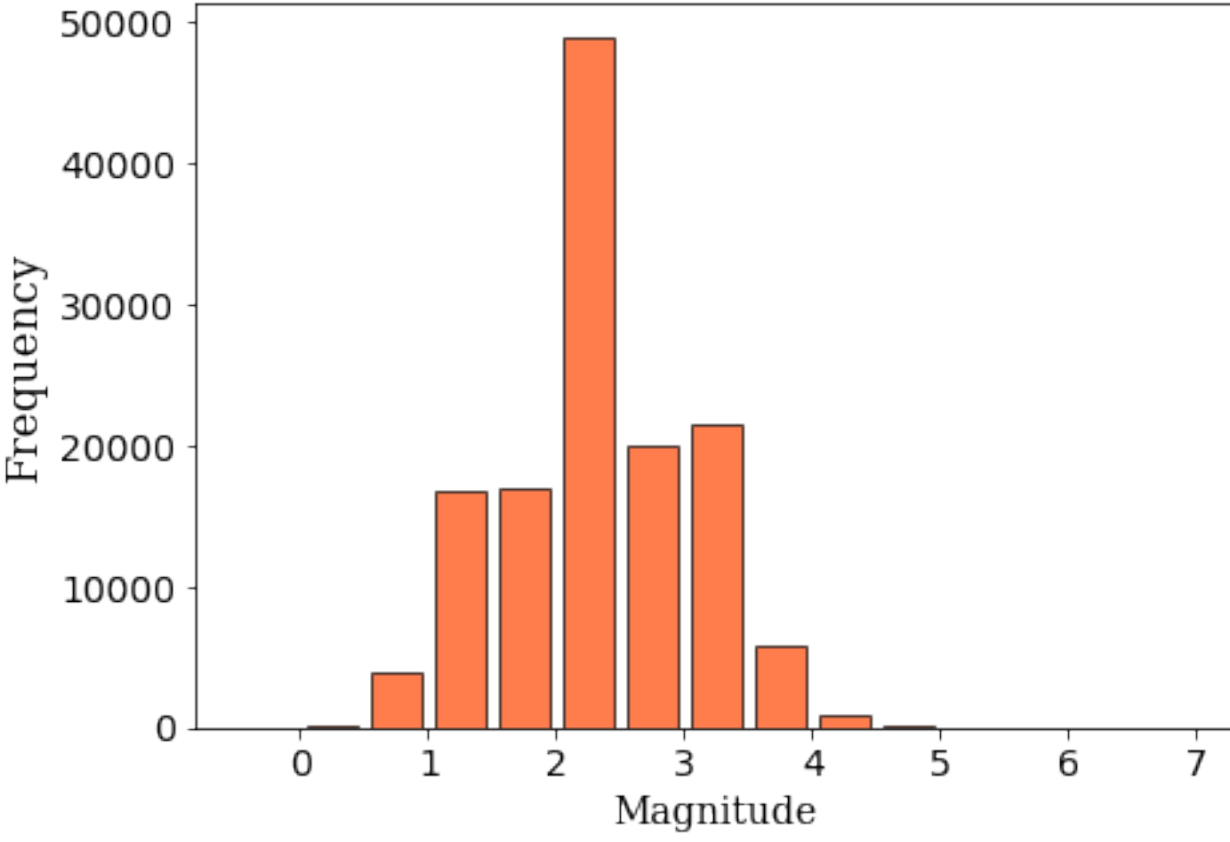
(a)


Error in predicted Magnitude
4
2
0
−2
−4
Error in predicted P-arrival time (seconds)
1.0
0.5
0.0
−0.5
−1.0
−1.5
Frequency
60000
40000
20000
0
20
40
60
80
100
120
SNR of data (dB)

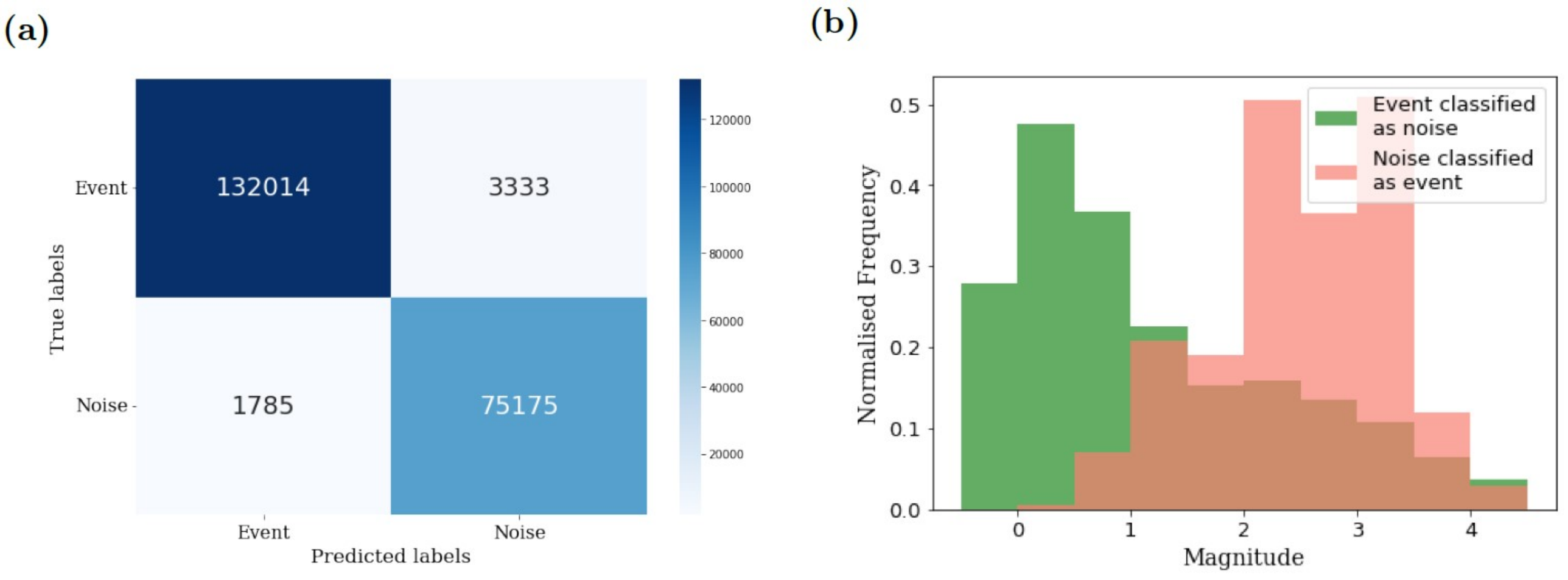
(b)


Error in predicted Magnitude
4
2
0
−2
−4
Error in predicted P-arrival time (seconds)
1.0
0.5
0.0
−0.5
−1.0
−1.5
Frequency
20000
15000
10000
5000
0
0
50
100
150
200
250
300
350
400
Hypocentral distance (km)

FigureB1.

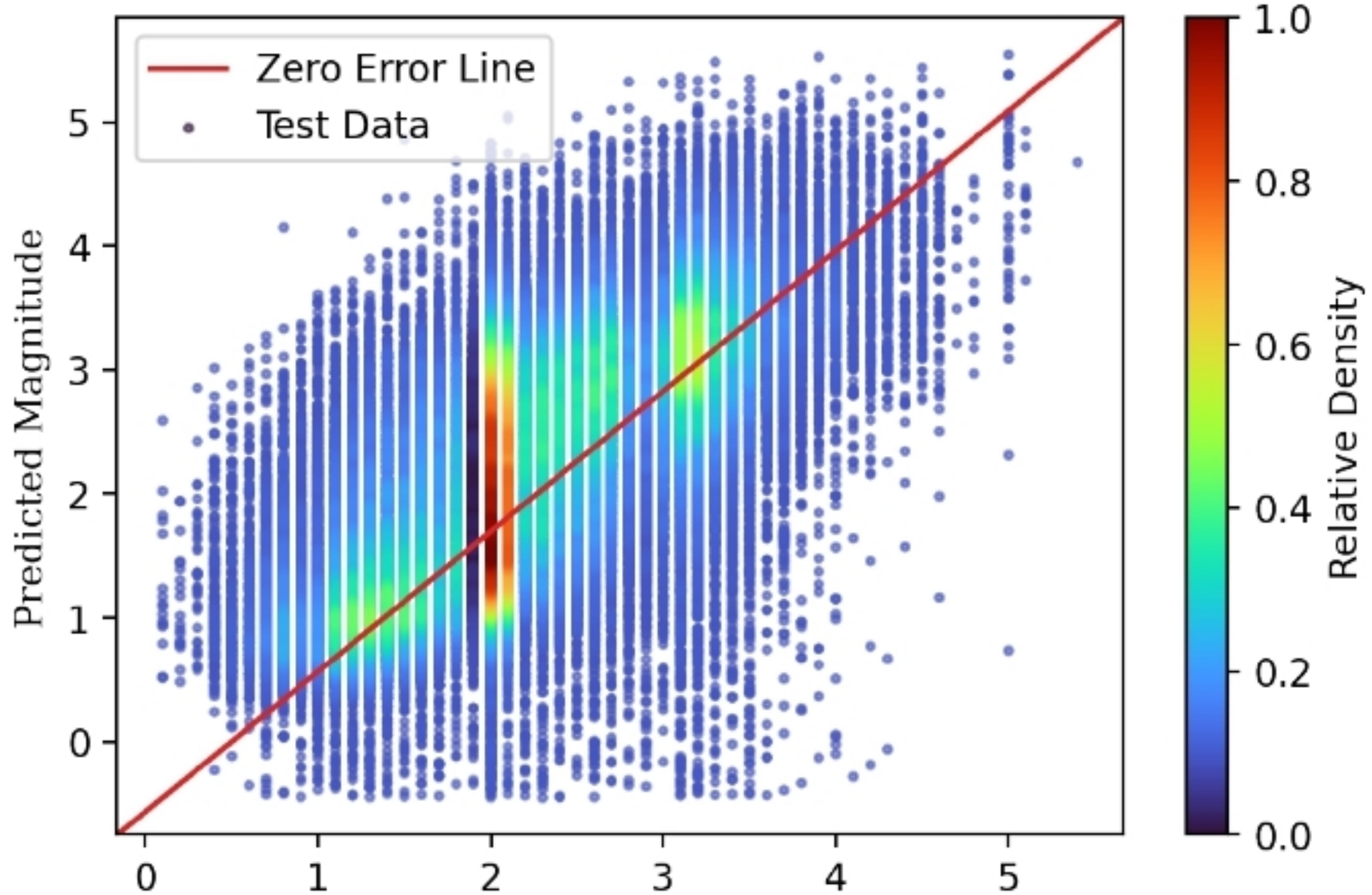

50000
40000
30000
20000
10000
0
Frequency
0
1
2
3
4
5
6
7
Magnitude

FigureB2.

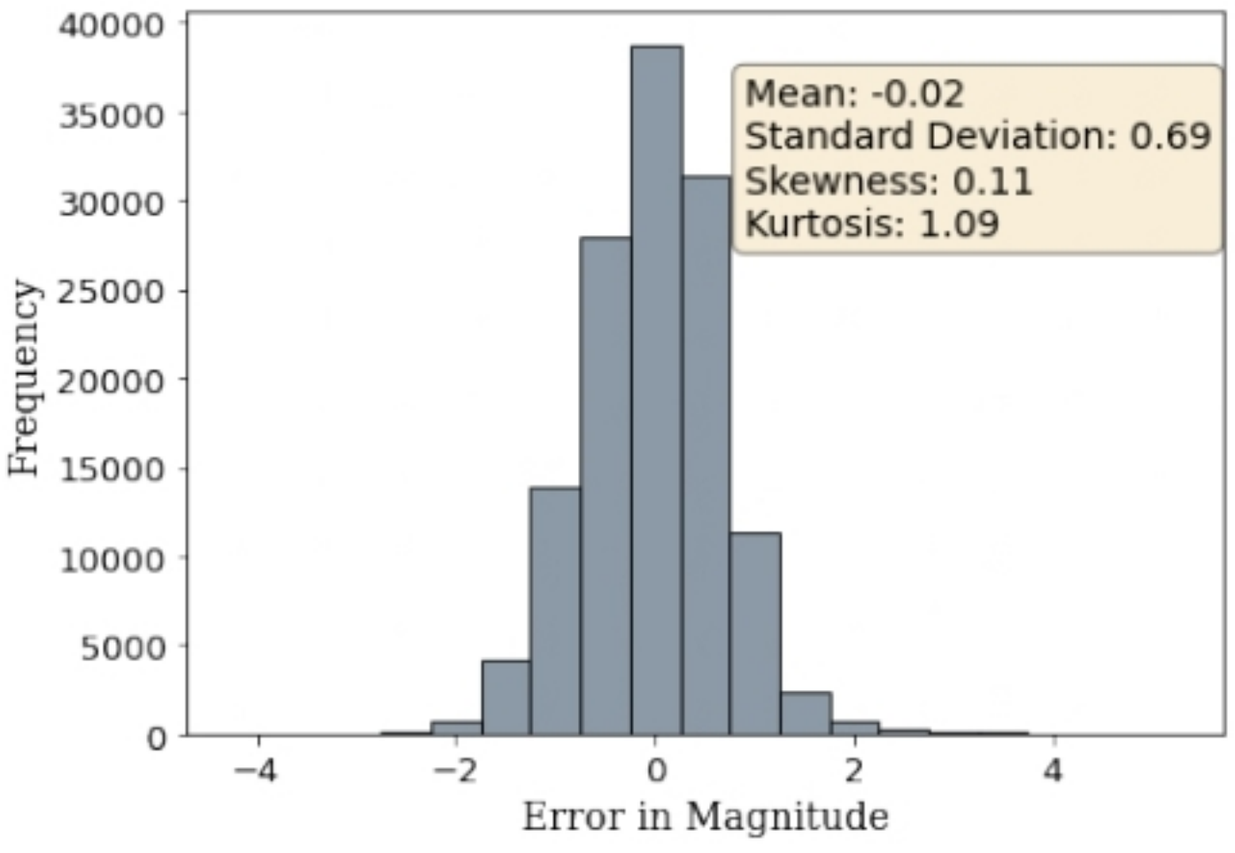

(a)
132014
3333
1785
75175
Event
Noise
True labels
Event
Noise
Predicted labels
120000
100000
80000
60000
40000
20000
(b)
Event classified as noise
Noise classified as event
Normalised Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
Magnitude

FigureB3.

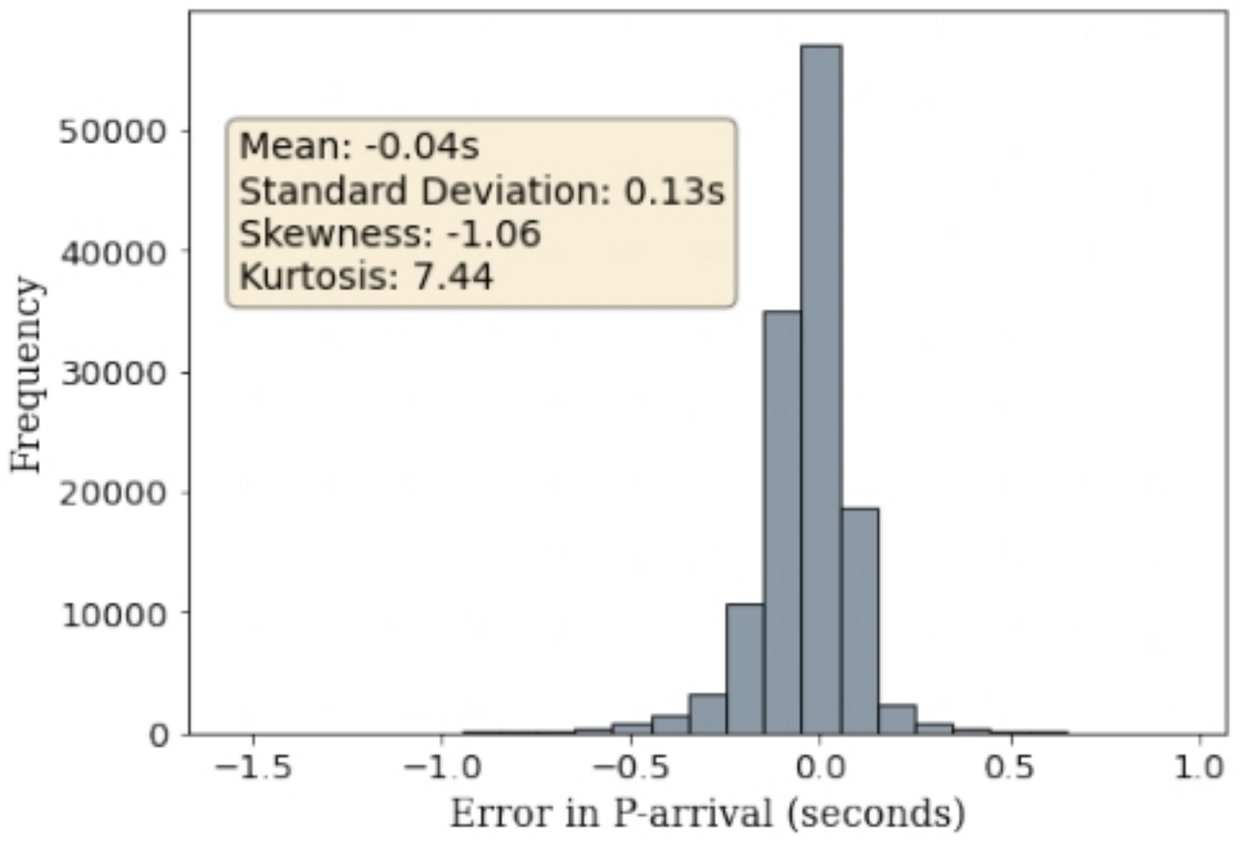
(a)


Zero Error Line
Test Data
Predicted Magnitude
True Magnitude
Relative Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0

(b)

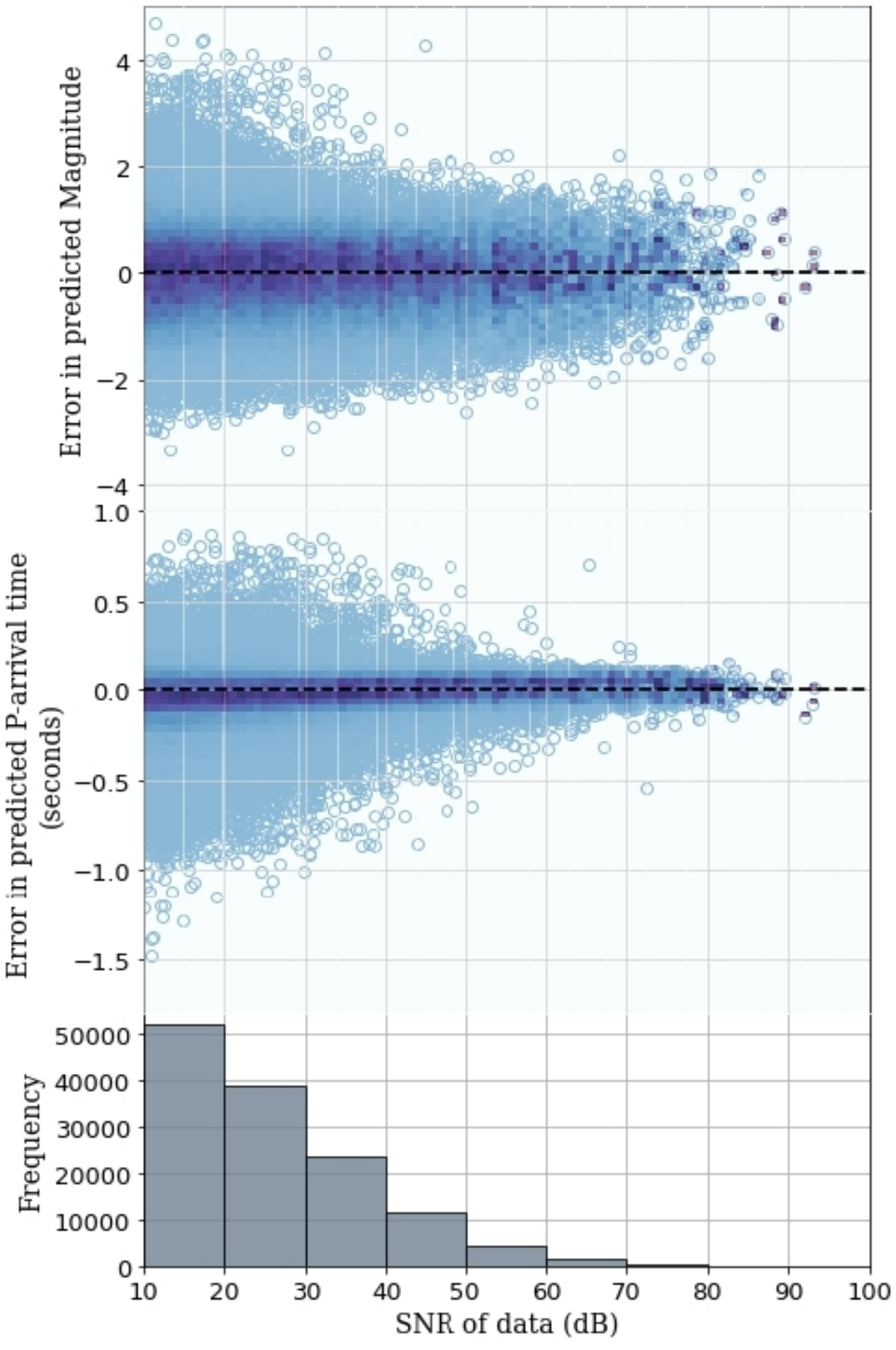

Mean: -0.02
Standard Deviation: 0.69
Skewness: 0.11
Kurtosis: 1.09
Frequency
Error in Magnitude
0
5000
10000
15000
20000
25000
30000
35000
40000
−4
−2
0
2
4

(c)

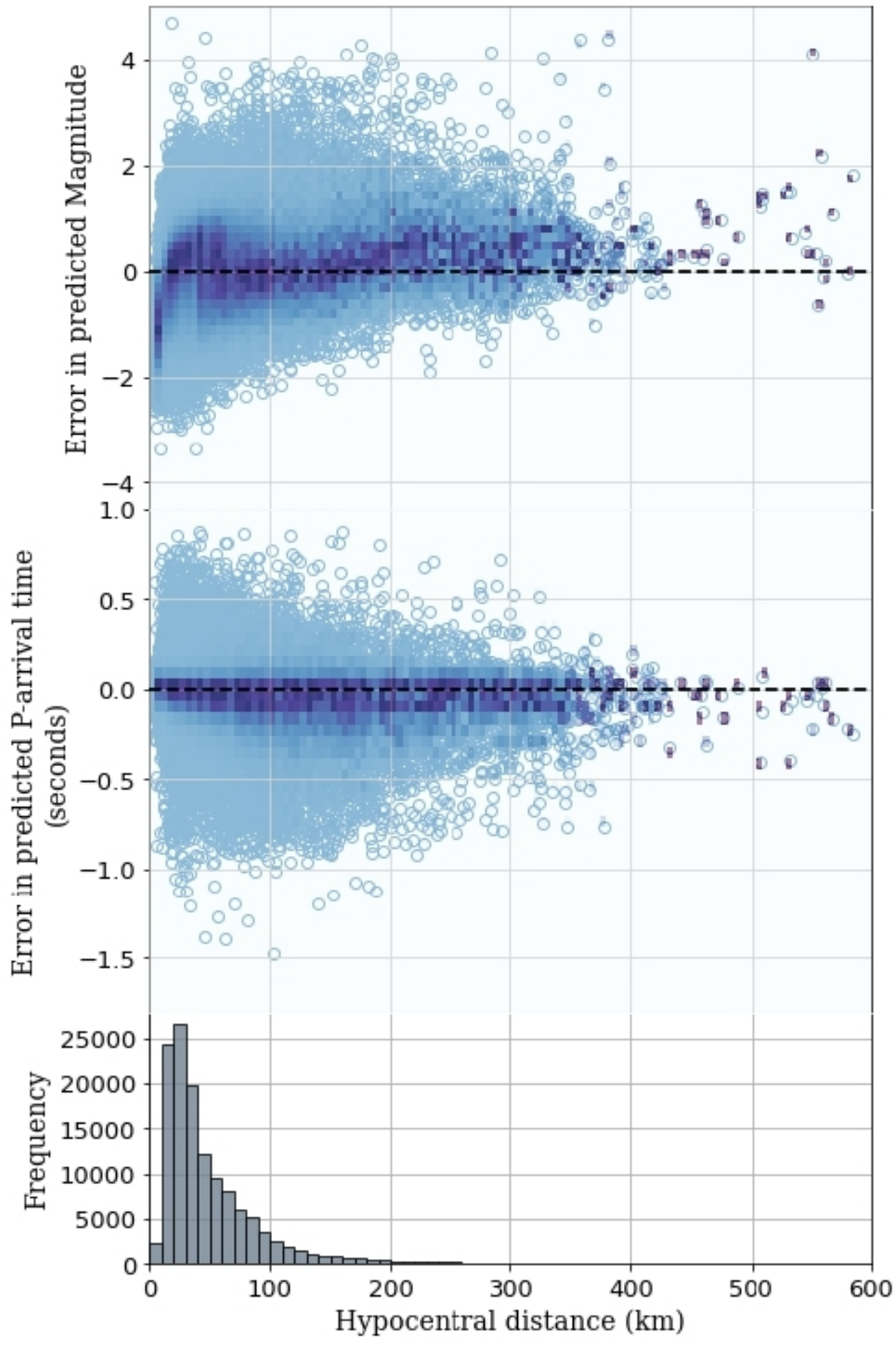

Mean: -0.04s
Standard Deviation: 0.13s
Skewness: -1.06
Kurtosis: 7.44
Frequency
Error in P-arrival (seconds)
0
10000
20000
30000
40000
50000
−1.5
−1.0
−0.5
0.0
0.5
1.0

FigureB4.

(a)


Error in predicted Magnitude
4
2
0
−2
−4
Error in predicted P-arrival time (seconds)
1.0
0.5
0.0
−0.5
−1.0
−1.5
Frequency
50000
40000
30000
20000
10000
0
10
20
30
40
50
60
70
80
90
100
SNR of data (dB)

(b)


Error in predicted Magnitude
4
2
0
−2
−4
Error in predicted P-arrival time (seconds)
1.0
0.5
0.0
−0.5
−1.0
−1.5
Frequency
25000
20000
15000
10000
5000
0
0
100
200
300
400
500
600
Hypocentral distance (km)